LỜI NÓI ĐẦU

Trong những năm gần đây, sự phát triển của Tin học đã làm thay đổi nhiều ngành truyền thống của Lí thuyết số (trong cuốn sách này, chúng ta thường dùng từ "Số học"). Nếu như trước thập kỷ 70, số học vẫn được xem là một trong những ngành lí thuyết xa rời thực tiễn nhất, thì ngày nay, nhiều thành tựu mới nhất của số học có ứng dụng trực tiếp vào các vấn đề của đời sống, như thông tin, mật mã, kĩ thuật máy tính. Một phương hướng mới của số học ra đời và phát triển mạnh mẽ: số học thuật toán. Có thể nói, đó là chiếc cầu nối giữa số học với tin học. Với việc sử dụng rộng rãi máy tính trong nghiên cứu số học, nhiều người cho rằng, số học ngày nay đã thành một khoa học thực nghiệm! Điều đó thể hiện khá rõ trong những "thuật toán xác suất" được đề cập đến trong cuốn sách này.

Mục đích của cuốn sách nhỏ này là cung cấp cho người đọc một số kiến thức sơ bộ về số học thuật toán. Cuốn sách không đòi hỏi ở người đọc một kiến thức chuẩn bị nào về lý thuyết số. Vì thế cũng có thể gọi nó là "Nhập môn thuật toán vào số học". Điều đó có nghĩa là, trong nhiều con đường khác nhau để đi vào số học, ta chọn con đường thuật toán: các định lí, khái niệm của số học được trình bày cùng với các thuật toán xây dựng chúng. Trong nhiều trường hợp, các thuật toán có kèm theo đánh giá sơ bộ về độ phức tạp.

Cuốn sách nhằm một số đối tượng khá rộng rãi: những sinh viên, nghiên cứu sinh về số học và tin học, những người quan tâm đến lí thuyết và ứng dụng của số học hiện đại. Nhiều phần của cuốn sách có thể có ích cho học sinh các lớp chuyên toán và chuyên tin học.

Chương đầu tiên của cuốn sách được dành để giới thiệu vài định nghĩa cơ bản nhất của lí thuyết thuật toán. Ba chương tiếp theo trình bày những vấn đề cơ sở của số học. Chương 5, ngoài việc chuẩn bị kiến thức cho những phần tiếp theo, có bình luận ít nhiều về vai trò của sự tương tự giữa số và đa thức trong sự phát triển của số học hiện đại.

Để người đọc có thể hình dung phần nào các ứng dụng của số học thuật toán, cuốn sách dành chương 6 để nói về lí thuyết mật mã. Một vài ứng dụng gần đây của lí thuyết đường cong elliptic vào mật mã được trình bày trong chương 7. Cũng có thể xem Chương 7 là một nhập môn ngắn và sơ cấp vào lí thuyết đường cong elliptic, một trong những lí thuyết phong phú nhất của Hình học đại số số học.

Cuối mỗi chương đều có một số bài tập dành cho độc giả muốn đọc cuốn sách "một cách tích cực". Một số bài tập mang tính chất luyện tập và tính toán thực hành, một số khác là mở rộng lí thuyết. Trừ chương cuối về đường cong elliptic, các chương còn lại đều có kèm theo hướng dẫn thực hành tính toán bằng chương trình MAPLE. Phần hướng dẫn thực hành này do Tạ Thị Hoài An biên soạn. Cuối cuốn sách có phần tự kiểm tra kiến thức dành cho những độc giả học giáo trình này với sự trợ giúp của máy tính.

Do nhiều nguyên nhân khác nhau, cuốn sách chắc chắn còn rất nhiều thiếu sót. Tác giả hy vọng nhận được những lời phê bình của bạn đọc.

Hà nội, 1998
Hà Huy Khoái

Chương 1.

THUẬT TOÁN

§1. Định nghĩa.

Có thể định nghĩa thuật toán theo nhiều cách khác nhau. Ở đây chúng tôi không có ý định trình bày chặt chẽ về thuật toán như trong một giáo trình logic, mà sẽ hiểu khái niệm *thuật toán* theo một cách thông thường nhất.

Thuật toán là một qui tắc để, với những dữ liệu ban đầu đã cho, tìm được lời giải sau một khoảng thời gian hữu hạn.

Để minh hoạ cách ghi một thuật toán, cũng như tìm hiểu các yêu cầu đề ra cho thuật toán, ta xét trên các ví dụ cụ thể sau đây.

Cho n số $X[1], X[2],..., X[n]$, ta cần tìm m và j sao cho $m=X[j] = \max_{1\le k\le n} X[k]$, và j là lớn nhất có thể. Điều đó có nghĩa là cần tìm cực đại của các số đã cho, và chỉ số lớn nhất trong các số đạt cực đại.

Với mục tiêu tìm số cực đại với chỉ số lớn nhất, ta xuất phát từ giá trị $X[n]$. Bước thứ nhất, vì mới chỉ có một số, ta có thể tạm thời xem $m=X[n]$ và $j=n$. Tiếp theo , ta so sánh $X[n]$ với $X[n-1]$. Trong trường hợp n-1=0, tức n=1, thuật toán kết thúc.

Nếu $X[n-1] \le X[n]$, ta chuyển sang so sánh $X[n]$ với $X[n-2]$ .Trong trường hợp ngược lại, $X[n-1]$ chính là số cực đại trong hai số đã xét, và ta phải thay đổi m và j: đặt $m= X[n-1], j=n-1$. Với cách làm như trên, ở mỗi bước, ta luôn nhận được số cực đại trong những số đã xét. Bước tiếp theo là so sánh nó với những số đứng trước, hoặc kết thúc thuật toán trong trường hợp không còn số nào đứng trước nó.

Thuật toán mô tả trên đây được ghi lại như sau:

Thuật toán tìm cực đại.

M1. [Bước xuất phát] Đặt j←n, k←n-1, m← X[n].

M2. [Đã kiểm tra xong?] Nếu k=0, thuật toán kết thúc.

M3. [So sánh] Nếu X[k]≤m,chuyển sang M5.

M4. [Thay đổi m] Đặt j ← k, m ← X[k]. (Tạm thời m đang là cực đại)

M5. [Giảm k] Đặt k ← k-1, quay về M2.

Dấu "←" dùng để chỉ một phép toán quan trọng là phép thay chỗ (replacement).

Trên đây ta ghi một thuật toán bằng ngôn ngữ thông thường. Trong trường hợp thuật toán được viết bằng ngôn ngữ của máy tính, ta có một *chương trình*.

Trong thuật toán có những số liệu ban đầu, được cho trước khi thuật toán bắt đầu làm việc: các *đầu vào (*input). Trong thuật toán M, đầu vào là các số *X[1], X[2]*,..., *X[n]*.

Một thuật toán có thể có một hoặc nhiều *đầu ra* (ouput). Trong thuật toán M, các đầu ra là *m* và *j*.

Có thể thấy rằng thuật toán vừa mô tả thoả mãn các yêu cầu của một thuật toán nói chung, đó là:

1. *Tính hữu hạn*.Thuật toán cần phải kết thúc sau một số hữu hạn bước. Khi thuật toán ngừng làm việc, ta phải thu được câu trả lời cho vấn đề đặt ra. Thuật toán M rõ ràng thoả mãn điều kiện này, vì ở mỗi bước, ta luôn chuyển từ việc xét một số sang số đứng trước nó, và số các số là hữu hạn.

2. *Tính xác định*. Ở mỗi bước, thuật toán cần phải xác định, nghĩa là chỉ rõ việc cần làm. Nếu đối với người đọc, thuật toán M chưa thoả mãn điều kiện này thì đó là lỗi của người viết!

Ngoài những yếu tố kể trên, ta còn phải xét đến tính hiệu quả của thuật toán. Có rất nhiều thuật toán, về mặt lí thuyết là kết thúc sau hữu hạn bước, tuy nhiên thời gian "hữu hạn" đó vượt quá khả năng làm việc của chúng ta. Những thuật toán đó sẽ không được xét đến ở đây, vì chúng ta chỉ quan tâm những thuật toán có thể sử dụng thật sự trên máy tính.

Cũng do mục tiêu nói trên, ta còn phải chú ý đến *độ phức tạp* của các thuật toán. Độ phức tạp của một thuật toán có thể đo bằng *không gian*, tức là dung lượng bộ nhớ của máy tính cần thiết để thực hiện thuật toán, và bằng *thời gian*, tức là thời gian máy tính làm việc. Trong cuốn sách này, khi nói đến độ phức tạp của thuật toán, ta luôn hiểu là độ phức tạp thời gian.

§2. Độ phức tạp thuật toán.

Dĩ nhiên, thời gian làm việc của máy tính khi chạy một thuật toán nào đó không chỉ phụ thuộc vào thuật toán, mà còn phụ thuộc vào máy tính được sử dụng. Vì thế, để có một tiêu chuẩn chung, ta sẽ đo độ phức tạp của một thuật toán bằng số các phép tính phải làm khi thực hiện thuật toán. Khi tiến hành cùng một thuật toán, số các phép tính phải thực hiện còn phụ thuộc vào cỡ của bài toán, tức là độ lớn của đầu vào. Vì thế, độ phức tạp của thuật toán sẽ là một hàm số của độ lớn của đầu vào. Trong những ứng dụng thực tiễn, chúng ta không cần biết chính xác hàm này, mà chỉ cần biết " cỡ" của chúng, tức là cần có một ước lượng đủ tốt của chúng.

Khi làm việc, máy tính thường ghi các chữ số bằng những bóng đèn "sáng, tắt": bóng đèn sáng chỉ số 1, bóng đèn tắt chỉ số 0. Vì thế thuận tiện nhất là dùng hệ đếm cơ số 2, trong đó để biểu diễn một số, ta chỉ cần dùng hai kí hiệu 0 và 1. Một kí hiệu 0 hoặc 1 được gọi là một *bit* (viết tắt của chữ "binary digit"). Một số nguyên *n* biểu diễn bởi *k* chữ số 1 và 0 dược gọi là một *số k-bit*. Trong chương tiếp theo, ta sẽ thấy rằng, số tự nhiên *n* sẽ là một số *k*-bit với $k=[\log_2 n]$ (dấu[ ] kí hiệu phần nguyên của một số).

Độ phức tạp của một thuật toán được đo bằng số các *phép tính bit*. Phép tính bit là một phép tính logic hay số học thực hiện trên các số 1-bit 0 và 1.

Để ước lượng độ phức tạp của thuật toán, ta dùng khái niệm *bậc O-lớn*.

Định nghĩa 1.1: Giả sử $f(n)$ và $g(n)$ là hai hàm xác định trên tập hợp các số nguyên dương. Ta nói *$f(n)$ có bậc O-lớn của $g(n)$,* và viết $f(n)=O(g(n))$ hoặc $f=O(g)$, nếu tồn tại một số $C>0$ sao cho với n đủ lớn, các hàm $f(n)$ và $g(n)$ đều dương, đồng thời $f(n) < Cg(n)$.

Ví dụ. 1) Giả sử $f(n)$ là đa thức;

$$f(n)=a_d n^d + a_{d-1} n^{d-1} + ...+a_1 n+a_0,$$

trong đó $a_d > 0$. Dễ chứng minh rằng $f(n)=O(n^d)$.

2) Nếu $f_1(n)=O(g(n)), f_2(n)=O(g(n))$ thì $f_1+f_2=O(g)$.

3) Nếu $f_1=O(g_1), f_2=O(g_2)$, thì $f_1.f_2=O(g_1.g_2)$.

4) Nếu tồn tại giới hạn hữu hạn

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

thì $f=O(g)$.

5)Với mọi số $\varepsilon>0$, $\log n=O(n^{\varepsilon})$.

Định nghĩa 1.2. Một thuật toán được gọi là có *độ phức tạp đa thức*, hoặc *có thời gian đa thức*, nếu số các phép tính cần thiết khi thực hiện thuật toán không vượt quá $O(\log^d n)$, trong đó n là độ lớn của đầu vào, và d là số nguyên dương nào đó.

Nói cách khác, nếu đầu vào là các số k-bit thì thời gian thực hiện thuật toán là $O(k^d)$, tức là tương đương với một đa thức của k.

Các thuật toán với thời gian $O(n^{\alpha})$, $\alpha>0$, được gọi là các thuật toán với *độ phức tạp mũ*, hoặc *thời gian mũ*.

Chú ý rằng, nếu một thuật toán nào đó có độ phức tạp $O(g)$, thì cũng có thể nói nó độ phức tạp $O(h)$ với mọi hàm $h > g$. Tuy nhiên, ta luôn luôn cố gắng tìm ước lượng tốt nhất có thể được để tránh hiểu sai về độ phức tạp thực sự của thuật toán.

Cũng có những thuật toán có độ phức tạp trung gian giữa đa thức và mũ. Ta thường gọi đó là thuật toán *dưới mũ*. Chẳng hạn, thuật toán nhanh nhất được biết hiện nay để phân tích một số nguyên n ra thừa số là thuật toán có độ phức tạp

$$\exp(\sqrt{\log n \log\log n}).$$

Khi giải một bài toán, không những ta chỉ cố gắng tìm ra một thuật toán nào đó, mà còn muốn tìm ra thuật toán " tốt nhất". Đánh giá độ phức tạp là một trong những cách để phân tích, so sánh và tìm ra thuật toán tối ưu. Tuy nhiên, độ phức tạp không phải là tiêu chuẩn duy nhất để đánh giá thuật toán. Có những thuật toán, về lí thuyết thì có độ phức tạp cao hơn một thuật toán khác, nhưng khi sử dụng lại có kết quả

(gần đúng) nhanh hơn nhiều. Điều này còn tuỳ thuộc những bài toán cụ thể, những mục tiêu cụ thể, và cả kinh nghiệm của người sử dụng.

Chúng ta cần lưu ý thêm một điểm sau đây. Mặc dù định nghĩa thuật toán mà chúng ta đưa ra chưa phải là chặt chẽ, nó vẫn quá "cứng nhắc " trong những ứng dụng thực tế! Bởi vậy, chúng ta còn cần đến các thuật toán "xác suất ", tức là các thuật toán phụ thuộc vào một hay nhiều tham số ngẫu nhiên. Những "thuật toán" này, về nguyên tắc không được gọi là thuật toán, vì chúng có thể, với xác suất rất bé, không bao giờ kết thúc. Tuy nhiên, thực nghiệm chỉ ra rằng, các thuật toán xác suất thường hữu hiệu hơn các thuật toán không xác suất. Thậm chí, trong rất nhiều trường hợp, chỉ có các thuật toán như thế là sử dụng được.

Khi làm việc với các thuật toán xác suất, ta thường hay phải sử dụng các số "ngẫu nhiên". Khái niệm chọn số ngẫu nhiên cũng cần được chính xác hoá. Thường thì người ta sử dụng một "máy" sản xuất số giả ngẫu nhiên nào đó. Tuy nhiên, trong cuốn sách này, chúng tôi không đề cập đến vấn đề nói trên, mà mỗi lần nói đến việc chọn số ngẫu nhiên, ta sẽ hiểu là điều đó thực hiện được trên máy.

Cũng cần lưu ý ngay rằng, đối với các thuật toán xác suất, không thể nói đến thời gian tuyệt đối, mà chỉ có thể nói đến thời gian hy vọng (expected).

Để hình dung được phần nào "độ phức tạp" của các thuật toán khi làm việc với những số lớn, ta xem bảng dưới đây cho khoảng thời gian cần thiết để phân tích một số nguyên n ra thừa số bằng thuật toán nhanh nhất được biết hiện nay (ta xem máy tính sử dụng vào việc này có tốc độ 1 triệu phép tính trong 1 giây)

Số chữ số thập phân	Số phép tính bit	Thời gian
50	$1{,}4.10^{10}$	3,9 giờ
75	$9{,}0.10^{12}$	104 ngày
100	$2{,}3.10^{15}$	74 năm
200	$1{,}2.10^{23}$	$3{,}8.10^{9}$ năm
300	$1{,}5.10^{29}$	$4{,}9.10^{15}$ năm
500	$1{,}3.10^{39}$	$4{,}2.10^{25}$ năm

Từ bảng trên đây, ta thấy rằng, ngay với một thuật toán dưới mũ, thời gian làm việc với các số nguyên lớn là quá lâu. Vì thế nói chung người ta luôn cố gắng tìm những thuật toán đa thức.

Lí thuyết về độ phức tạp thuật toán là một lí thuyết rất phong phú. Trong cuốn sách này, chúng tôi không lấy mục tiêu trình bày lí thuyết đó làm trọng tâm. Độc giả quan tâm đến lí thuyết thuật toán có thể tìm đọc các sách trong phần Tài liệu tham khảo.

Chương 2.

SỐ NGUYÊN

§1. Biểu diễn số nguyên và các phép tính số học

1.1 Hệ cơ số.

Mặc dù hầu hầu hết độc giả đã quen thuộc với cách biểu diễn số nguyên trong cơ số tuỳ ý, chúng tôi nhắc lại sơ qua vấn đề đó ở phần này, để thuận tiện cho việc trình bày các thuật toán về số nguyên.

Định lí 2.1. *Giả sử b là một số nguyên lớn hơn 1. Khi đó mọi số nguyên n có thể viết duy nhất dưới dạng*

$$n=a_k b^k + a_{k-1}b^{k-1} +...+ a_1 b^1 +a_0,$$

trong đó a_j *là số nguyên,* $0 \le a_j \le k-1$, *với* $j=0,1,...,k$ *và hệ số đầu tiên* $a_k \neq 0$.

Chứng minh. Ta chỉ cần thực hiện liên tiếp phép chia *n* cho *b*:

$$n=bq_0 +a_0,\ 0 \le a_0 \le b-1.$$

Nếu $q_0 > b$, ta tiếp tục chia q_0 cho *b* để được

$$q_0=bq_1 +a_1,\ 0 \le a_1 \le b-1.$$

Tiếp tục quá trình đó, ta có:

$$q_1=bq_2 +a_2,\ 0 \le a_2 \le b-1$$

$$q_2=bq_3 +a_3,\ 0 \le a_3 \le b-1$$

$$... ... ...$$

$$q_{k-1}=b.0 +a_k,\ 0 \le a_k \le b-1.$$

Quá trình kết thúc vì ta luôn có: $n > q_0 > q_1 > ... \ge 0$.

Chúng tôi dành cho độc giả việc chứng minh *n* có dạng như trong phát biểu của định lí, và biểu diễn đó là duy nhất.

Số *b* nói trong định lí được gọi là *cơ số* của biểu diễn. Các hệ biểu diễn cơ số 10 và 2 tương ứng được gọi là hệ thập phân và nhị phân. Các hệ số a_j được gọi là các chữ số. Về sau ta dùng bit để chỉ chữ số nhị phân.

Nếu số nguyên *n* biểu diễn trong cơ số *b* có *k* chữ số, thì từ chứng minh trên, ta có :

$$b^{k-1} \le n \le b^k.$$

Như vậy số chữ số của *n* được tính theo công thức:

$$k=[\log_b n]+1=[\log n / \log b]+1,$$

trong đó, kí hiệu log dùng để chỉ logarit cơ số e. Trong cơ số tuỳ ý, ta có: $k=O(\log n)$.

Để phân biệt các biểu diễn của số nguyên trong những hệ cơ số khác nhau, ta thường dùng cách viết $(a_k a_{k-1}...a_1 a_0)_b$ để chỉ số $n= a_k b^k + a_{k-1} b^{k-1} +...+ a_1 b^1 + a_0$,

Ví dụ. 1). Đối với số 1994 trong hệ thập phân, ta có $(1994)_{10}=(11111001010)_2$.

2). Trong máy tính, bên cạnh hệ cơ số 2, người ta cũng thường dùng hệ cơ số 8 hoặc 16. Lí do chủ yếu là vì việc chuyển một số viết ở cơ số này sang cơ số kia trong 3 cơ số đó được thực hiện một cách dễ dàng. Ví dụ, muốn chuyển một số cho trong cơ số 2 sang cơ số 8, ta chỉ việc nhóm từ phải sang trái từng khối 3 chữ số, rồi chuyển số được viết trong khối đó sang dạng thập phân. Chẳng hạn, số $(1110010100110)_2$ được tách thành các nhóm 1,110,010,100,110. Từ đó ta được:

$$(1110010100110)_2=(16246)_8.$$

Ta có thể làm tương tự để chuyển số đã cho thành số viết trong cơ số 16, chỉ cần nhóm thành từng bộ 4 chữ số. Chú ý rằng, trong trường hợp này, cần thêm vào các kí hiệu mới để chỉ các "chữ số " từ 10 đến 15.

Ta nhắc lại rằng máy tính sử dụng cách viết nhị phân, hoặc là các "bit". Máy tính nào cũng có giới hạn về độ lớn của các số có thể đưa vào tính toán. Giới hạn đó được gọi là *cỡ từ* của máy, kí hiệu qua w. Cỡ từ thường là một luỹ thừa của 2, chẳng hạn 2^{35}.

Để thực hiện các phép tính số học với những số nguyên lớn hơn cỡ từ, ta làm như sau. Muốn đưa một số $n > w$ vào máy, ta viết n dưới dạng cơ số w, và khi đó n được biểu diễn bằng những số không vượt quá cỡ từ. Ví dụ, nếu cỡ từ của máy là 2^{35}, thì để đưa vào một số có độ lớn cỡ 2^{350}-1, ta chỉ cần dùng 10 số nhỏ nhỏ hơn cỡ từ của máy, bằng cách biểu diễn n trong cơ số 2^{35}. Như đã nói trong ví dụ ở 1, việc chuyển một số từ cơ số 2 sang cơ số 2^{35} được thực hiện dễ dàng bằng cách nhóm từng khối 35 chữ số.

Từ qui tắc của các phép tính số học, ta thấy rằng:

1) Để cộng hoặc trừ hai số nguyên k bit, ta cần $O(k)$ phép tính bit.

2) Để nhân hoặc chia hai số k bit theo qui tắc thông thường, ta cần $O(k^2)$ phép tính bit.

Trong những thập kỉ gần đây, người ta tìm ra những thuật toán nhân với độ phức tạp bé hơn nhiều so với cách nhân thông thường. Điều thú vị là, nếu thoạt nhìn thì các thuật toán đó "phức tạp" hơn quy tắc nhân thông thường. Tuy nhiên, khi làm việc với những số rất lớn, các thuật toán này cho phép thực hiện việc nhân hai số với một thời gian bé hơn hẳn so với quy tắc thông thường.

1.2 Thuật toán nhân nhanh hai số.

Ta sử dụng tính chất hết sức đơn giản của phép nhân: nếu $a=a_1+a_2$, $b=b_1+b_2$, thì $ab=a_1b_1+a_2b_2+a_2b_1+a_1b_2$. Điều đáng chú ý ở đây là, thay cho việc nhân hai số nguyên n bit, ta thực hiện việc nhân các số có chữ số nhỏ hơn, cùng với một số phép

cộng (đòi hỏi số phép tính bit ít hơn là phép nhân). Thực ra điều này không có gì mới: ngay trong quan niệm ban đầu, phép nhân a với b đã là phép cộng b lần số a!.

Tuy nhiên để có một thuật toán nhân nhanh, ta không thể cộng b lần số a, mà phải tìm được một cách tối ưu nào đó để tách b và a thành những phần nhỏ hơn. Những thuật toán trình bày dưới đây cho chúng ta một số cách để làm việc phân chia như vậy.

Giả sử muốn nhân hai số nguyên $2n$ bit,

$$a=(a_{2n-1}a_{2n-2}...a_1a_0)_2,$$

$$b=(b_{2n-1}b_{2n-2}...b_1b_0)_2.$$

Ta viết $a=2^nA_1+A_0$, $b=2^nB_1+B_0$, trong đó

$$A_1=(a_{2n-1}a_{2n-2}...a_n)_2,\ A_0=(a_{n-1}a_{n-2}...a_1a_0)_2,$$

$$B_1=(b_{2n-1}b_{2n-2}...b_n)_2,\ B_0=(b_{n-1}b_{n-2}...b_1b_0)_2.$$

Khi đó ta có:

$$ab=(2^{2n}+2^n)A_1B_1+2^n(A_1 - A_0)+(2^n+1)A_0B_0. \qquad (1.1)$$

Như vậy, việc nhân hai số a,b $2n$ bit được đưa về việc nhân các số n bit, cùng với các phép cộng, trừ và dịch chuyển (nhân một số với một luỹ thừa bậc n của 2 được thực hiện bằng cách dịch số đó sang trái n vị trí).

Định lí 2.2. *Thuật toán 2.1 có độ phức tạp là* $O(n^{\log_2 3})$.

Chứng minh. Gọi $M(n)$ là số các phép tính bit tối đa cần thiết khi thực hiện nhân hai số nguyên n bit bằng thuật toán 2.1. Từ công thức (1.1) ta có:

$$M(2n)\le 3M(n)+Cn,$$

trong đó C là một hằng số không phụ thuộc n. Đặt $c=\max(C,M(2))$.

Bằng quy nạp, dễ chứng minh được rằng

$$M(2^k) \le c(3^k-2^k).$$

Từ đó ta có

$$M(n)=M(2^{\log_2 n}) \le M(2^{[\log_2 n]+1}) \le c(3^{[\log_2 n]+1}-2^{[\log_2 n]+1}) \le 3c.3^{[\log_2 n]} \le 3c.3^{\log_2 n}=3cn^{\log_2 3}.$$

Định lí đã được chứng minh.

Với thuật toán 2.1, ta thấy rằng, ngay chỉ với cách phân chia đơn giản số nguyên thành hai phần với số chữ số bằng nhau, ta đã nhận được một thuật toán giảm đáng kể thời gian thực hiện phép nhân. Dĩ nhiên, cách phân chia như vậy còn xa với cách phân chia tối ưu.

Ta sẽ chứng tỏ rằng cách phân chia như trên có thể tổng quát hoá để nhận được những thuật toán nhân với độ phức tạp nhỏ hơn nhiều.

Cũng như trước đây, ta sẽ kí hiệu qua $M(n)$ số các phép tính bit cần thiết để thực hiện phép nhân hai số nguyên n bit. Trước tiên, ta chứng minh công thức sau: với mọi số tự nhiên n, tồn tại thuật toán sao cho:

$$M((r+1)n) \le (2r+1)M(n)+Cn, \tag{1.2}$$

với C là một hằng số nào đó. Như vậy, Định lí 2.2 là trường hợp riêng với r=1.

Giả sử cần nhân hai số $(r+1)n$ bit:

$$a=(a_{(r+1)n-1}...a_1a_0)_2,$$

$$b=(b_{(r+1)n-1}...b_1b_0)_2.$$

Ta tách mỗi số a,b thành $r+1$ số hạng:

$$a=A_r2^{rn}+...+A_12^n+A_0$$

$$b=B_r2^{rn}+...+B_12^n+B_0,$$

trong đó A_j,B_j là các số n bit.

Ta nhận xét rằng, việc biểu diễn một số nguyên dưới dạng cơ số nào đó cũng gần giống như viết số đó dưới dạng đa thức, trong đó các chữ số chính là các hệ số của đa thức. Vì vậy việc nhân hai số có thể thực hiện tương tự như việc nhân đa thức. Ta xét các đa thức sau:

$$A(x)=A_rx^r+...+A_1x+A_0,$$

$$B(x)=B_rx^r+...+B_1x+B_0,$$

$$W(x)=A(x)B(x)=W_{2r}x^{2r}+...+W_1x+W_0.$$

Từ định nghĩa các đa thức trên ta được: $a=A(2^n),b$=B(2^n), $ab= W(2^n)$. Như vậy, ta dễ dàng tính được tích ab nếu biết được các hệ số của đa thức $W(x)$.

Công thức (1.2) sẽ được chứng minh nếu ta tìm được một thuật toán tính các hệ số của $W(x)$ mà chỉ sử dụng $2r+1$ phép nhân các số n bit và một số phép tính khác với độ phức tạp O(n). Điều đó có thể làm bằng cách tính giá trị của đa thức $W(x)$ tại $2r+1$ điểm sau đây:

$$W(0)=A(0)B(0),\ W(1)=A(1)B(1),...,\ W(2r)=A(2r)B(2r).$$

Chú ý rằng, các số A_j,B_j không nhất thiết là các số n bit, nhưng với r cố định, chúng có số chữ số nhiều nhất là r+t, với một t cố định nào đó. Dễ thấy rằng, có thể nhân hai số $(r+t)$-bit với không quá $M(n)+c_1n$ phép tính bit, trong đó c_1 là hằng số (chỉ cần tách số $(n+t)$-bit thành hai phần n-bit và t-bit, và nhận xét rằng, khi t cố định, việc nhân số t-bit với số n-bit đòi hỏi không quá cn phép tính bit).

Khi đã có các giá trị $W(j),(j$=0,1,...$2r)$, ta tìm được đa thức $W(x)$ theo công thức Lagrange:

$$W(x)=\sum_{j=0}^{2r}(-1)^j\mathrm{W(j)}\frac{\mathrm{x(x-1)...(x-j+1)(x-j-1)...(x-2r)}}{\mathrm{j!(2r-j)!}}\ .$$

Như vậy, các hệ số của $W(x)$ sẽ là tổ hợp tuyến tính (với hệ số không phụ thuộc n) của các giá trị $W(j)$, và do đó, tính được bằng $O(n)$ phép tính bit.

Ta đã chứng minh được công thức sau:

$$M((r+1)n) \le (2r+1)M(n)+Cn.$$

Lập luận tương tự như trong chứng minh định lí 2.1 ta có:

$$M(n) \le C_3 n^{\log_{r+1}(2r+1)} < C_3 n^{1+\log_{r+1}2}.$$

Với mọi $\varepsilon > 0$ bé tuỳ ý, khi các thừa số có số chữ số rất lớn, ta có thể chọn r đủ lớn sao cho $\log_{r+1}2 < \varepsilon$. Ta có định lí sau:

Định lí 2.3. *Với mọi $\varepsilon > 0$, tồn tại thuật toán nhân sao cho số phép tính bit $M(n)$ cần thiết để nhân hai số n bit thoả mãn bất đẳng thức*

$$M(n) < C(\varepsilon) n^{1+\varepsilon},$$

với hằng số $C(\varepsilon)$ nào đó độc lập với n.

Nhận xét. Có thể chứng minh được rằng, với cách chọn r "đủ tốt", ta có thuật toán nhân hai số n-bit sao cho

$$M(n) = O(n \log_2 n \log\log_2 n).$$

Chứng minh định lí đó không khó, nhưng khá dài (xem [Kr]).

§2. Số nguyên tố.

Định nghĩa 2.4. Số *nguyên tố* là số nguyên lớn hơn 1, không chia hết cho số nguyên dương nào ngoài 1 và chính nó. Số nguyên lớn hơn 1 không phải là số nguyên tố được gọi là *hợp số*.

Dễ chứng minh được rằng, số các số nguyên tố là vô hạn (Bài tập 2.14).

Như ta sẽ thấy trong những chương tiếp theo, bài toán xác định một số cho trước có phải là số nguyên tố hay không có nhiều ứng dụng trong thực tiễn. Đối với những số nhỏ, bài toán đó dĩ nhiên không có gì khó. Tuy nhiên, khi làm việc với những số lớn, ta cần phải tìm ra những thuật toán hữu hiệu, nghĩa là có thể thực hiện được trên máy tính trong một khoảng thời gian chấp nhận được. Khi nói đến "những số lớn", ta thường hiểu là những số nguyên dương có khoảng 100 chữ số thập phân trở lên.

Để có thể tìm ra những thuật toán xác định nhanh một số có phải là số nguyên tố hay không, ta cần hiểu sâu sắc tính chất các số nguyên tố. Trong chương này, ta chỉ đi vào các tính chất cơ bản nhất.

Định lí sau đây cho một thuật toán đơn giản để xác định các số nguyên tố.

Định lí 2.5. *Mọi hợp số n đều có ước nguyên tố nhỏ hơn $\sqrt{n}$.*

Thật vậy, vì n là một hợp số nên ta có thể viết $n=ab$, trong đó a và b là các số nguyên với $1<a\le b<n$. Rõ ràng ta phải có a hoặc b không vượt quá $\sqrt{n}$, giả sử đó là a. Ước nguyên tố của a cũng đồng thời là ước nguyên tố của n.

Từ định lí trên, ta có thuật toán sau đây để tìm ra các số nguyên tố nhỏ hơn hoặc bằng số n cho trước.

Sàng Eratosthenes. Trước tiên, ta viết dãy các số tự nhiên từ 1 đến n. Trong dãy đó gạch đi số 1, vì nó không phải là số nguyên tố. Số nguyên tố đầu tiên của dãy là 2. Tiếp theo đó ta gạch khỏi dãy số tất cả những số chia hết cho 2. Số đầu tiên không chia hết cho 2 là 3: đó chính là số nguyên tố. Ta lại gạch khỏi dãy còn lại những số nào chia hết cho 3. Tiếp tục như thế, ta gạch khỏi dãy những số chia hết cho mọi số nguyên tố bé hơn $\sqrt{n}$. Theo định lí trên, những số còn lại của dãy là tất cả các số nguyên tố không vượt quá n. Thật vậy, các hợp số không vượt quá n, theo định lí trên, đều phải có ước nguyên tố nhỏ hơn $\sqrt{n}$, và do đó đã bị gạch khỏi dãy số trong một bước nào đó của thuật toán.

Sàng Eratosthenes, mặc dù cho ta thuật toán xác định mọi số nguyên tố không vượt quá một số cho trước, rất ít được sử dụng để xác định xem một số đã cho có phải là số nguyên tố hay không. Nguyên nhân là vì thuật toán có độ phức tạp quá lớn: để kiểm tra n, ta phải thực hiện phép chia cho tất cả các số nguyên tố không vượt quá $\sqrt{n}$.

Ta hãy xét sơ qua về độ phức tạp của thuật toán nói trên. Với mỗi số thực dương x cho trước ta kí hiệu $\pi(x)$ số các số nguyên tố không vượt quá x. Khi đó, theo định lí Hadamard-Valée-Poussin ta có:

$$\lim_{x\to\infty} \pi(x) / \frac{x}{\log x} = 1.$$

Như vậy, số các số nguyên tố không vượt quá $\sqrt{n}$ là vào khoảng $\sqrt{n}/\log\sqrt{n} = 2\sqrt{n}/\log n$. Để chia n cho m, ta cần $O(\log_2 n.\ \log_2 m)$ phép tính bit. Như vậy, số các phép tính bit cần thiết để kiểm tra n có phải là số nguyên tố hay không ít nhất là $(2\sqrt{n}/\log n)(C\log_2 n) = C\sqrt{n}$ (ở đây ta dùng ước lượng rất sơ lược $\log_2 m \ge 1$). Như vậy, nếu n vào cỡ khoảng 100 chữ số thập phân, số các phép tính bit phải dùng sẽ vào cỡ 10^{50}. Với những máy tính thực hiện một triệu phép tính trong một giây, thời gian cần thiết sẽ vào khoảng $3{,}1.10^{36}$ năm!

Ta kết thúc tiết này bằng định lý quan trọng sau đây, thường được gọi là *định lý cơ bản của số học* .

Định lí 2.6. *Mọi số nguyên tố lớn hơn 1 đều phân tích được một cách duy nhất thành tích các số nguyên tố, trong đó các thừa số được viết với thứ tự không giảm.*

Chứng minh. Giả sử tồn tại những số không viết được thành tích các số nguyên tố. Gọi n là số bé nhất trong các số đó. Như vậy, n phải là hợp số, $n=a.b$, với $a, b<n$. Do định nghĩa của n các số a và b phân tích được thành tích các số nguyên tố, nghĩa là n cũng phân tích được. Mâu thuẫn với giả thiết.

Còn phải chứng minh phân tích là duy nhất. Giả sử ta có:

$$n=p_1 p_2 ...p_s=q_1q_2...q_r,$$

trong đó p_i, q_j là các số nguyên tố. Giản ước những số nguyên tố bằng nhau có mặt trong hai vế, ta được đẳng thức

$$p_{i1}p_{i2}...p_{iu}=q_{j1}q_{j2}...q_{jv},,$$

trong đó không có số nguyên tố nào có mặt cả hai vế. Như vậy, vế trái chia hết cho q_{j1}, và do đó phải tồn tại một thừa số của tích chia hết cho q_{j1}: điều đó vô lý, vì đây là tích các số nguyên tố khác với q_{j1}.

Phân tích như trên của các số nguyên được gọi là phân tích ra thừa số nguyên tố. Khi n là một số rất lớn, việc kiểm tra xem n là số nguyên tố hay hợp số, và nếu là hợp số thì tìm phân tích của nó ra thừa số nguyên tố, là một bài toán hết sức khó khăn. Trong những phần tiếp theo của cuốn sách, ta sẽ tìm hiểu nhiều thuật toán để làm việc đó, cũng như các ứng dụng của nó trong thực tiễn.

§3. Thuật toán Euclid.

Một trong những thuật toán cơ bản và lâu đời nhất của toán học là thuật toán Euclid. Thuật toán đó cho phép xác định ước chung lớn nhất của hai số nguyên cho trước.

Khi trình bày thuật toán Euclid, ta nhắc lại sơ qua khái niệm đồng dư. Những tính chất cần dùng của đồng dư và các tính chất cơ bản của ước chung lớn nhất được cho trong các bài tập của chương này.

Giả sử m là một số nguyên dương. Ta nói hai số nguyên a và b là đồng dư với nhau modulo m nếu m chia hết hiệu a-b (ta dùng cách viết $m \mid (a-b)$). Để chỉ quan hệ đồng dư, ta dùng ký hiệu $a \equiv b \pmod m$.

Như vậy, $a \equiv b \pmod m$ khi và chỉ khi tồn tại số nguyên k sao cho $a=b+km$.

Quan hệ đồng dư là một trong những quan hệ cơ bản của số học, và ta sẽ gặp thường xuyên trong những phần tiếp theo của cuốn sách. Trong thuật toán Euclid, ta chỉ dùng quan hệ đó để diễn đạt ngắn gọn về phần dư của phép chia.

Thuật toán sau đây cho phép tính ước chung lớn nhất (ƯCLN) d của hai số nguyên không âm a và b (ký hiệu là $d=(a,b)$).

Thuật toán Euclid

E1. [Kết thúc?] Nếu b=0, in ra a và kết thúc thuật toán.

E2. [Chia Euclid] Đặt r←a mod b, a←b, b←r và quay về bước 1.

Ví dụ: tính d=(24,63) bằng thuật toán Euclid.

Ta có: d=(24,63) = (15,24)=(9,15)=(6,9)=(3,6)=(0,3)=3.

Định lý sau đây vừa cho ta một chứng minh tính đúng đắn của thuật toán Euclid, vừa cho một ước lượng về độ phức tạp của nó.

Định lí Lamé. *Số phép chia cần thiết để tìm ƯCLN của hai số nguyên dương bằng thuật toán Euclid không vượt quá 5 lần số chữ số thập phân của số bé trong hai số đã cho.*

Chứng minh. Giả sử $a>b$ là hai số nguyên dương cho trước. Bằng thuật toán Euclid, ta có: $a=r_0$, $b=r_1$ và:

$$r_0=r_1q_1+r_2,\ 0\le r_2<r_1$$

$$r_1=r_2q_2+r_3,\ 0\le r_3<r_2$$

$$\dots\dots\dots\dots$$

$$r_{n-2}=r_{n-1}q_{n-1}+r_n,\ 0\le r_n<r_{n-1}$$

$$r_{n-1}=r_nq_n$$

Như vậy, ta đã làm n phép chia. Trong các phép chia đó, ta có: $q_1, q_2, \dots, q_{n-1} \ge 1$, $q_n \ge 2$, $r_n<r_{n-1}$. Từ đó suy ra:

$$r_n \ge 1=f_2,$$

$$r_{n-1} \ge 2r_n \ge 2f_2=f_3$$

$$r_{n-2} \ge r_{n-1}+ r_n \ge f_3+f_2=f_4$$

$$r_{n-3} \ge r_{n-2}+ r_{n-1} \ge f_4+f_5=f_6$$

$$\dots\dots\dots\dots$$

$$r_2 \ge r_3+ r_4 \ge f_{n-1}+f_{n-2}=f_n$$

$$b=r_1 \ge r_2+ r_3 \ge f_n+f_{n-1}=f_{n+1}$$

Chú ý rằng, dãy số $\{f_n\}$ nhận được chính là dãy số Fibonaci quen thuộc trong số học. Đối với dãy số này, bằng quy nạp, dễ chứng minh ước lượng sau đây:

$$f_n>\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}.$$

Từ bất đẳng thức $b \ge f_{n+1}$ ta có:

$$\log_{10}b \ge (n-1)\log_{10}\left(\frac{1+\sqrt{5}}{2}\right)>(n-1)/5$$

Định lí được chứng minh.

Hệ quả 2.6. *Giả sử $a<b$, khi đó số các phép tính bit cần thiết để thực hiện thuật toán Euclid là $O((\log_2 a)^3)$.*

Thật vậy, số phép chia phải làm là $O(\log_2 a)$, và mỗi phép chia cần $O((\log_2 a)^2)$ phép tính bit.

Thuật toán Euclid, mặc dù đã ra đời hàng nghìn năm, vẫn là thuật toán tốt nhất để tìm ƯCLN của hai số nguyên cho trước! Cho đến năm 1967, J.Stein xây dựng được một thuật toán khá thuận tiện để tìm ƯCLN trong trường hợp các số đã cho được viết dưới dạng nhị phân. Ưu điểm chủ yếu của thuật toán này là ta không cần làm các phép tính chia (thực ra ta có làm phép chia số chẵn cho 2, nhưng trong cơ số 2 thì đó là phép dịch chuyển số đã cho sang phải một vị trí). Thuật toán dựa trên những nhận xét đơn giản sau (xem phần bài tập cuối chương):

1) Nếu a,b là các số chẵn, thì $(a,b)=2(a/2,b/2)$.

2) Nếu a chẵn, b lẻ, thì $(a,b)=(a/2,b)$.

3) Nếu a,b đều lẻ thì $a-b$ chẵn và $|a-b|<\max(a,b)$.

4) $(a,b)=(a-b,b)$.

Thuật toán đó được mô tả như sau (chúng tôi dành phần chứng minh cho độc giả).

Thuật toán tìm ƯCLN của hai số nguyên dương a,b.

E'1. (Tìm luỹ thừa của 2) Đặt $k\leftarrow 0$ và lập liên tiếp phép tính sau cho đến khi ít nhất một trong hai số a, b lẻ: đặt $k\leftarrow k+1$, $a\leftarrow a/2$, $b\leftarrow b/2$.

E'2. (Xuất phát). (ở bước xuất phát này, a, b đều đã được chia cho 2^k, và có ít nhất một trong hai số là lẻ). Nếu a lẻ, đặt $t\leftarrow -b$ và chuyển sang E'4. Nếu ngược lại, đặt $t\leftarrow a$.

E'3. (Chia đôi t). (Tại thời điểm này, t chẵn, khác 0). Đặt $t\leftarrow t/2$.

E'4. (t có chẵn hay không?) Nếu t chẵn quay về E'3.

E'5. (Sắp xếp lại max(a,b)). Nếu $t>0$, đặt $a\leftarrow t$; nếu ngược lại, đặt $b\leftarrow -t$. Như vậy, số lớn nhất trong hai số đã được thay bởi $|t|$.

E'6. (Trừ) Đặt $t\leftarrow a-b$. Nếu $t\neq 0$, quay lại E'3. Nếu ngược lại thuật toán kết thúc và in ra $a.2^k$.

Ngoài thuật toán Euclid nói trên, trong nhiều trường hợp, ta cần đến thuật toán Euclid mở rộng. Thuật toán này không những cho ta thuật toán tìm ƯCLN của hai số a, b, mà còn cho ta biểu diễn $d=(a,b)$ dưới dạng tổ hợp tuyến tính của a, b: $d=ma+nb$, trong đó m, n là các số nguyên.

Trước hết, ta chứng minh bổ đề sau:

Bổ đề 2.7: *ƯCLN của các số nguyên a và b là số d dương nhỏ nhất biểu diễn được dưới dạng tổ hợp tuyến tính của a và b.*

Thật vậy, giả sử d là số nguyên dương nhỏ nhất biểu diễn được dưới dạng $d=ma+nb$. Ta chứng tỏ d là ước chung của a và b. Xét phép chia $a=dq+r$, trong đó $0\le r<d$. Rõ ràng r cũng là một tổ hợp tuyến tính của a và b, nên do d là số nguyên dương nhỏ nhất có tính chất đó, $r=0$. Tương tự, d là ước của b. Dễ thấy rằng, mọi ước chung khác của a và b cũng là ước của d: vậy d chính là ước chung lớn nhất.

Khi cho hai số a, b, để tìm biểu diễn của d như trong bổ đề, ta thường là như sau: viết a=$bv+q$, $0\le q<b$. Sau đó,lại viết $b=uq+r=u(a-bv)+r$, $0\le r<q$. Tiếp tục quá trình đó, do các số dư q, r giảm dần nên ta thu được biểu diễn cần thiết. Điều vừa nói được thể hiện trong thuật toán sau đây, mà chứng minh chặt chẽ được dành cho độc giả.

Thuật toán Euclid mở rộng.

Cho hai số nguyên không âm u, v, tìm (u_1,u_2,u_3) sao cho $(u,v)=u_3=uu_1+vu_2$. Trong tính toán, ta thêm vào các ẩn phụ (v_1,v_2,v_3), (t_1,t_2,t_3) và luôn có trong mọi bước các đẳng thức sau đây:

$$ut_1+vt_2=t_3,\ uv_1+vv_2=v_3,\ uu_1+vu_2=u_3.$$

Ed1.(Xuất phát). Đặt $(u_1,u_2,u_3)\leftarrow(1,0,u)$, $(v_1,v_2,v_3)\leftarrow(0,1,v)$.

Ed2. (Kiểm tra $v_3=0$?) Nếu $v_3=0$, thuật toán kết thúc.

Ed3. (Chia, trừ). Đặt $q\leftarrow[u_3/v_3]$, và sau đó đặt $(t_1,t_2,t_3)\leftarrow(u_1,u_2,u_3)-q(v_1,v_2,v_3)$, $(v_1,v_2,v_3)\leftarrow(t_1,t_2,t_3)$ và quay về bước 2.

Ví dụ. Cho $a=63$, $b=24$. Dùng thuật toán Euclid ta có:

- Bước 1. $u_1=1$, $u_2=0$, $u_3=63$, $v_1=0$, $v_2=1$, $v_3=24$.

- Bước 2. $q=2$, $u_1=0$, $u_2=1$, $u_3=24$, $v_1=1$, $v_2=-2$, $v_3=15$.

- Bước 3. $q=1$, $u_1=1$, $u_2=-2$, $u_3=15$, $v_1=-1$, $v_2=3$, $v_3=9$.

- Bước 4. $q=1$, $u_1=-1$, $u_2=3$, $u_3=9$, $v_1=2$, $v_2=-5$, $v_3=6$.

- Bước 5. $q=1$, $u_1=2$, $u_2=-5$, $u_3=6$, $v_1=-3$, $v_2=8$, $v_3=3$.

- Bước 6. $q=2$, $u_1=-3$, $u_2=8$, $u_3=3$, $v_1=8$, $v_2=-21$, $v_3=0$.

Ta có biểu diễn: 3=(-3)64+8.24.

§4. Định lí Trung Quốc về phần dư:

Giả sử $m_1,m_2,\ldots,m_r$ là các số nguyên dương nguyên tố cùng nhau từng cặp. Khi đó hệ đồng dư:

$$x_1\equiv a_1(\mathrm{mod}\ m_1),$$

$$x_2\equiv a_2(\mathrm{mod}\ m_2),$$

... ...

$$x_r \equiv a_r (\mathrm{mod}\ m_r).$$

Có nghiệm duy nhất modulo $M=m_1m_2...m_r$.

Chứng minh. Trước hết ta xây dựng một nghiệm của hệ.

Giả sử $M_k=M/m_k= m_1m_2...m_{k-1}m_{k+1}...m_r$. Ta biết rằng $(M_k,m_k)=1$ vì $(m_j,m_k)=1$ với mọi $j \neq k$. Như vậy, theo bài tập 2.18 ta có thể tìm một nghịch đảo y_k của M_k modulo m_k, tức là $M_k y_k \equiv 1 \pmod{m_k}$.

Đặt

$$x=a_1M_1y_1+ a_2M_2y_2 +...+ a_rM_ry_r .$$

Ta thấy rằng $x \equiv a_k (\mathrm{mod}\ m_k)$ với mọi k vì $m_k | M_j$ với $j \neq k$ nên $M_j \equiv 0 \pmod{m_k}$ khi $j \neq k$. Như vậy, x chính là một nghiệm của hệ đang xét.

Ta chứng tỏ rằng nghiệm vừa xây dựng là duy nhất modulo M. Giả sử x_0, x_1 là hai nghiệm của hệ. Khi đó, với mỗi k, $x_0 \equiv x_1 \equiv a_k \pmod{m_k}$, cho nên $m_k | (x_0-x_1)$. Theo bài tập 2.17, $M | (x_0-x_1)$. Định lí được chứng minh.

Định lí Trung Quốc về phần dư liên quan bài toán nổi tiếng "Hàn Tín điểm binh". Tương truyền rằng, để kiểm tra quân số, Hàn Tín thường ra lệnh cho quân sĩ xếp thành hàng 3, hàng 5, hàng 7 và thông báo cho ông các số dư. Khi biết các số dư và đã có sẵn thông tin gần đúng về số quân của mình, Hàn Tín dùng định lí trên đây để suy ra số quân chính xác.

Định lí Trung Quốc về phần dư được sử dụng trong máy tính để làm việc với những số lớn. Để đưa một số nguyên lớn tuỳ ý vào máy tính và làm các phép tính số học với chúng, ta cần có những kĩ thuật đặc biệt. Theo định lí Trung quốc về phần dư, khi cho trước các modun nguyên tố cùng nhau $m_1,m_2,...,m_r$, một số dương $n<M= m_1m_2...m_r$ được xác định duy nhất bởi các thặng dư dương bé nhất của nó theo modulo m_j với $j=1,2,...,r$. Giả sử rằng cỡ từ của máy chỉ là 100, nhưng ta cần làm các phép tính số học với những số nguyên cỡ 10^6. Trước tiên ta tìm các số nguyên nhỏ hơn 100, nguyên tố cùng nhau từng cặp, sao cho tích của chúng vượt quá 10^6. Chẳng hạn, ta có thể lấy $m_1=99$, $m_2=98$, $m_3=97$, $m_4=95$. Ta chuyển các số nguyên bé hơn 10^6 thành những bộ 4 số theo thặng dư dương bé nhất modulo m_1,m_2,m_3,m_4 (để làm được điều này, ta cũng phải làm việc với những số nguyên lớn! Tuy nhiên điều đó chỉ cần làm một lần với input, và một lần nữa với ouput). Như vậy, chẳng hạn để cộng các số nguyên, ta chỉ cần cộng các thặng dư dương bé nhất của chúng modulo m_1,m_2,m_3,m_4. Sau đó lại dùng định lí Trung Quốc về phần dư để tìm bộ 4 số tương ứng với tổng.

Ví dụ. Ta muốn tính tổng $x=123684$, $y=413456$ với một máy tính cỡ từ là 100. Ta có:

$$x \equiv 33(\mathrm{mod}\ 99), 8(\mathrm{mod}\ 98), 9(\mathrm{mod}\ 97), 89(\mathrm{mod}\ 95)$$

$$y \equiv 32(\mathrm{mod}\ 99), 92(\mathrm{mod}\ 98), 42(\mathrm{mod}\ 97), 16(\mathrm{mod}\ 95)$$

Như vậy,

$$x+y \equiv 65(\mathrm{mod}\ 99), 2(\mathrm{mod}\ 98), 51(\mathrm{mod}\ 97), 10(\mathrm{mod}\ 95)$$

Bây giời ta dùng định lí Trung Quốc về phần dư để tìm $x+y$ modulo M=99.98.97.95=89403930. Ta có: $M_1=M/99=903070$, $M_2=M/98=912288$, M_3=M/97=921690, $M_4=M/95=941094$. Ta cần tìm ngược của $M_i(\text{mod } y_i)$ với i=1,2,3,4, tức là giải hệ phương trình đồng dư sau đây (Bằng thuật chia Euclid):

$$903070y_1 \equiv 91y_1 \equiv 1(\text{mod } 99)$$

$$912285y_2 \equiv 3y_2 \equiv 1(\text{mod } 98)$$

$$921690y_3 \equiv 93y_3 \equiv 1(\text{mod } 97)$$

Ta tìm được: $y_1 \equiv 37(\text{mod } 99)$, $y_2 \equiv 38(\text{mod } 98)$, $y_3 \equiv 24(\text{mod } 97)$, $y_4 \equiv 4(\text{mod } 95)$.

Như vậy,

$$x+y=65.903070.37+2.912285.33+51.921690.24+10.941094.4=3397886480$$

$$\equiv 537140(\text{mod } 89403930)$$

Vì $0<x+y<89403930$, ta suy ra $x+y=537140$.

Rất có thể độc giả cho rằng, cách cộng hai số sử dụng định lí Trung Quốc về phần dư quá phức tạp so với cách cộng thông thường. Tuy nhiên, cần chú ý rằng, trong ví dụ trên đây, ta làm việc với các số nhỏ. Khi các số cần cộng có độ lớn vượt xa cỡ từ của máy, các quy tắc cộng "thông thường" không còn áp dụng được nữa.

Nói chung cỡ từ của máy tính là luỹ thừa rất lớn của 2, chẳng hạn 2^{35}. Như vậy, để sử dụng định lí Trung Quốc về phần dư, ta cần các số nhỏ hơn 2^{35} nguyên tố cùng nhau từng cặp. Để tìm các số nguyên như vậy, thuận tiện nhất là dùng các số dạng 2^m-1, trong đó m là số nguyên dương. Các phép tính số học với những số có dạng như vậy tương đối đơn giản dựa vào bổ đề sau.

Bổ đề 2.8. *Nếu a và b là các số nguyên dương thì thặng dư dương bé nhất modulo 2^b-1 của 2^a-1 là 2^r-1, trong đó r là thặng dư dương bé nhất của a modulo b.*

Thật vậy, nếu $a=bq+r$, trong đó r là thặng dư dương bé nhất của a modulo b, thì ta có

$$(2^a-1)=(2^{bq+r}-1)=(2^b-1)(2b(q-1)+r+...+2^{b+r}+2^r)+(2^r-1).$$

Hệ quả 2.9. *Nếu a và b là các số nguyên dương, thì ước chung lớn nhất của 2^a-1 và 2^b-1 là $2^{(a,b)}$-1.*

Hệ quả 2.10. *Các số nguyên 2^a-1 và 2^b-1 nguyên tố cùng nhau khi và chỉ khi a và b nguyên tố cùng nhau.*

Chúng tôi dành việc chứng minh hai bổ đề này cho độc giả.

Ta có thể sử dụng hệ quả trên đây để tìm các số nhỏ hơn 2^{35}, nguyên tố cùng nhau từng cặp, sao cho tích của chúng lớn hơn một số đã cho. Giả sử ta cần làm các phép tính số học với những số nguyên có cỡ 2^{184}. Ta đặt: $m_1=2^{35}$-1, $m_2=2^{34}$-1, $m_3=2^{33}$-1, $m_4=2^{31}$-1, $m_5=2^{29}$-1, $m_6=2^{23}$-1. Vì số mũ của 2 trong các số trên nguyên tố với nhau từng cặp, nên theo hệ quả trên, các số đã chọn cũng nguyên tố với nhsu từng cặp. Ta có tích $m_1\ m_2\ m_3\ m_4\ m_5\ m_6>2^{184}$. Bây giờ ta có thể làm các phép tính số học với những số cỡ đến 2^{184}.

Trong các máy tính hiện đại, việc thực hiện nhiều phép tính được tiến hành đồng thời. Vì thế việc sử dụng định lí Trung Quốc về phần dư như trên lại càng tiện lợi: thay cho việc làm các phép tính với các số nguyên lớn, ta làm nhiều phép tính đồng thời với những số nguyên bé hơn. Điều đó giảm đáng kể thời gian tính toán.

Thuật toán giải phương trình đồng dư bằng định lí Trung Quốc

Từ chứng minh định lí Trung Quốc về phần dư, ta có thuật toán sau đây để giải hệ phương trình đồng dư $x \equiv x_i \pmod{m_i}$, trong đó m_i, $1 \le i \le k$ là các số nguyên tố với nhau từng cặp, x_i là các số nguyên cho trước. Trong thuật toán trình bày sau đây, chúng ta đã tìm ra cách để tránh phải làm việc với các số lớn như M_i và a_iM_i.

Thuật toán.

1. (Xuất phát). Đặt $j \leftarrow 2$, $C_1 \leftarrow 1$. Hơn nữa ta sắp xếp lại các số m_i theo thứ tự tăng dần.

2. (Tính toán sơ bộ). Đặt $p \leftarrow m_1m_2...m_{j-1} \pmod{m_j}$. Tính (u,v,d) sao cho $up+vm_j=d=UCLN(p,m_j)$ bằng thuật toán Euclid mở rộng.

Ed. Nếu d>0, in ra thông báo: các m_i không nguyên tố cùng nhau từng cặp. Nếu ngược lại, đặt $C_j \leftarrow u$, $j \leftarrow j+1$ và chuyển sang bước 3 nếu $j \le k$.

3. (Tính các hằng số phụ). Đặt $y_1 \leftarrow x_1 \bmod m_1$, và mỗi $j=2,...,k$ tính:

$y_j \leftarrow (x_j-(y_1+m_1(y_2+m_2(y_3+...+m_{j-2}y_{j-1})...))C_j \bmod m_j$.

4. (Kết thúc). In ra

$x \leftarrow y_1+m_1(y_2+m_2(y_3+...+m_{k-1}y_k)...)$, và kết thúc thuật toán.

§5. Một số đồng dư đặc biệt.

Định lí Wilson. *p là số nguyên tố khi và chỉ khi* $(p-1)! \equiv -1 \pmod{p}$.

Chứng minh. Trước tiên, giả sử *p* là số nguyên tố. Khi p=2, ta có $(p-1)! \equiv 1 \equiv -1 \pmod 2$. Bây giờ giả sử *p* là số nguyên tố lớn hơn 2. Theo bài tập 2.18, với mỗi số nguyên *a* với $1 \le a \le p-1$, tồn tại nghịch đảo $\bar{a}$, $1 \le \bar{a} \le p-1$, với $a\bar{a} \equiv 1 \pmod p$. Theo bài tập 2.13, trong số các số nguyên dương nhỏ hơn *p*, chỉ có 1 và *p*-1 là nghịch đảo với chính nó. Như vậy ta có thể nhóm các số nguyên từ 2 đến *p*-2 thành (*p*-3)/2 cặp số nguyên, tích của mỗi cặp đồng dư với 1 modulo *p*. Như vậy ta có:

$$2.3.....(p-3)(p-2) \equiv 1 \pmod p$$

Nhân hai vế với 1 và *p*-1 ta được:

$$(p-1)! \equiv 1.2.3...(p-2)(p-1) \equiv 1(p-1) \equiv -1 \pmod p$$

Ngược lại giả sử *p* thoả mãn đồng dư phát biểu trong định lí và *a* là một ước số của *p*, $a<p$. Khi đó, $a \mid (p-1)!$. Nhưng theo giả thiết, $p \mid (p-1)!+1$, từ đó suy ra $a=1$, vì là ước chung của *p* và (*p*-1)!. Vậy *p* là số nguyên tố, định lí được chứng minh.

Định lí Wilson có thể được dùng để kiểm tra một số có phải là số nguyên tố hay không. Tuy nhiên , dễ thấy rằng, thuật toán dựa theo định lí Wilson khó có thể sử dụng với những số nguyên lớn, bởi vì số các phép tính bit đòi hỏi quá cao.

Để đơn giản, ta gọi công việc xem xét một số đã cho có phải là số nguyên tố hay không là *kiểm tra nguyên tố*. Định lí sau đây có nhiều ứng dụng trong kiểm tra nguyên tố.

Định lí Fermat bé. *Nếu p là số nguyên tố và a là số không chia hết cho p thì* $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$.

Chứng minh. Xét p-1 số nguyên a, $2a,..., (p-1)a$. Các số đó đều không chia hết cho p và không có hai số nào đồng dư modulo p. Như vậy, các thặng dư dương bé nhất của chúng phải là 1, 2,... p-1, xếp theo thứ tự nào đó. Từ đó ta có:

$$a.2a.....(p-1) \equiv 1...(p-1) \equiv (p-1)!(\mathrm{mod}\ p)$$

tức là

$$a^{p-1}(p-1)! \equiv 1(\mathrm{mod}\ p)$$

Vì $((p-1)!,p)=1$ nên ta có $a^{p-1} \equiv 1(\mathrm{mod}\ p)$.

Hệ quả 2.11. *Nếu p là số nguyên tố và a là số nguyên dương thì* $a^p \equiv a (\mathrm{mod}\ p)$.

Hệ quả 2.12. *Nếu p là số nguyên tố và a là số nguyên không chia hết cho p thì* a^{p-2} *là nghịch đảo của a modulo p*.

Hệ quả 2.13. *Nếu a và b là các số nguyên dương, p nguyên tố, p/a thì các nghiệm của đồng dư thức tuyến tính* $ax \equiv b(mod\ p)$ *là các số nguyên x sao cho* $x \equiv a^{p-2}b(mod\ p)$.

§6. Số giả nguyên tố.

Theo định lí Fermat, nếu n là số nguyên tố và b là số nguyên tuỳ ý, thì $b^n \equiv b(\mathrm{mod}\ n)$. Do đó nếu tồn tại số b sao cho $b^n \not\equiv b(\mathrm{mod}\ n)$ thì n phải là hợp số. Trong nhiều ứng dụng , chúng ta lại cần đến các thuật toán để chỉ ra một số n là số nguyên tố. Trong trường hợp này, ta không thể dùng định lí Fermat bé, vì định lí ngược của nó không đúng. Tuy nhiên, nếu một số nguyên thoả mãn các giả thiết của định lí Fermat bé thì "có nhiều khả năng" nó là một số nguyên tố! Ta có định nghĩa sau đây.

Định nghĩa 2.14. Giả sử b là một số nguyên dương. Nếu n là hợp số nguyên dương và $b^n \equiv b(\mathrm{mod}\ n)$ thì n được gọi là *số giả nguyên tố cơ sở b*.

Trong trường hợp $(n,b)=1$, ta thường dùng định nghĩa tương đương: $b^{n-1} \equiv b(\mathrm{mod}\ n)$.

Ví dụ. Số nguyên 561=3.11.17 là số giả nguyên tố cơ sở 2. Thật vậy, áp dụng định lí Fermat bé, ta có $2^{560}=(2^2)^{280} \equiv 1(\mathrm{mod}\ 3)$, $2^{560}=(2^{10})^{56} \equiv 1(\mathrm{mod}\ 11)$, $2^{560}=(2^{16})^{35} \equiv 1(\mathrm{mod}\ 17)$. Từ đó suy ra (bài tập 2.12) $2^{560} \equiv 1(\mathrm{mod}\ 561)$.

Nói chung các số giả nguyên tố ít hơn nhiều so với các số nguyên tố. Chẳng hạn, có tất cả 4550525112 số nguyên tố bé hơn 10^{10}, nhưng chỉ có 14884 số giả nguyên tố cơ sở 2 trong khoảng đó. Sự kiện này giải thích cách nói ở trên: Các số thoả mãn định lí Fermat bé có nhiều khả năng là số nguyên tố. Tuy nhiên đối với mọi cơ sở tuỳ ý, số các số giả nguyên tố là vô hạn. Chẳng hạn, ta chứng minh điều đó đối với cơ sở 2.

Định lí 2.15. *Có vô số số giả nguyên tố cơ sở 2.*

Chứng minh. Giả sử n là một số giả nguyên tố cơ sở 2, ta sẽ chứng tỏ rằng, $m=2^n-1$ cũng là số giả nguyên tố cơ sở 2. Theo giả thiết, n là hợp số, chẳng hạn $n=dt$ $(1<d,t<n)$, và $2^{n-1}\equiv 1 (\mathrm{mod}\ n)$. Dễ thấy rằng m là hợp số, vì $(2^d-1) \mid (2^n-1)=m$. Do n là giả nguyên tố, tồn tại k sao cho $2^n-2=kn$. Ta có $2^{m-1}=2^{kn}$, và do đó, $m=(2^n-1)\mid(2^{nk}-1)=2^{m-1}-1$, tức là $2^{m-1}\equiv 1 (\mathrm{mod}\ m)$. Vậy số m là giả nguyên tố cơ sở 2.

Như vậy, để kiểm tra một số có phải là số nguyên tố hay không, trước tiên ta xem nó có là giả nguyên tố cơ sở 2 hay không, sau đó có thể tiếp tục kiểm tra đối với các cơ sở khác. Tuy nhiên, tồn tại các số giả nguyên tố với mọi cơ sở, đó là các số Carmichael.

Định nghĩa 2.16. Hợp số nguyên n thoả mãn $b^{n-1}\equiv 1 (\mathrm{mod}\ n)$ với mọi số nguyên dương b sao cho $(n,b)=1$ được gọi là *số Carmichael*.

Ví dụ. Số nguyên $561=3.11.17$ là một số Carmichael. Thật vậy, nếu $(b,561)=1$ thì $(b,3)=(b,11)=(b,17)=1$. Theo định lí Fermat bé, ta có $b^2\equiv 1 (\mathrm{mod}\ 3)$, $b^{10}\equiv 1 (\mathrm{mod}\ 11)$, $b^{16}\equiv 1 (\mathrm{mod}\ 17)$. Do đó, viết $560=2.280=10.56=16.35$ ta được:

$$b^{560}=(b^2)^{280}\equiv 1 (\mathrm{mod}\ 3),$$

$$b^{560}=(b^{10})^{56}\equiv 1 (\mathrm{mod}\ 11),$$

$$b^{560}=(b^{16})^{35}\equiv 1 (\mathrm{mod}\ 17).$$

Từ đó suy ra (bài tập 2.12): $b^{560}\equiv 1 (\mathrm{mod}\ 561)$.

Giả thuyết sau đây mới được chứng minh rất gần đây ([AGP]): tồn tại vô hạn số Carmichael.

Định lí sau đây cho một cách tìm số Carmichael.

Định lí 2.17. Nếu $n=q_1q_2...q_k$, *trong đó* q_j *là các số nguyên tố khác nhau thoả mãn* $(q_j-1) \mid (n-1)$, *thì n là số Carmichael.*

Thật vậy, giả sử b là số nguyên dương, $(b,n)=1$. Khi đó, $(b,q_j)=1$ với mọi j, và $b^{q_j-1}\equiv 1(\mathrm{mod}\, q_j)$. Vì $(q_j-1) \mid (n-1)$ nên $b^{n-1}\equiv 1 (\mathrm{mod}\ q_j)$, và do đó, $b^{n-1}\equiv 1 (\mathrm{mod}\ n)$.

Phần đảo của định lí trên đây cũng đúng, tuy nhiên được chứng minh hơi dài nên ta sẽ bỏ qua. Độc giả nào quan tâm có thể tìm đọc trong [Ro].

Như vậy, việc kiểm tra nguyên tố sẽ khó khăn khi gặp phải các số Carmicheal. Tuy nhiên, ta có thể khắc phục bằng cách sau đây. Nếu gặp đồng dư $b^{n-1}\equiv 1 (\mathrm{mod}\ n)$, ta chuyển sang xét đồng dư $b^{(n-1)/2}\equiv x (\mathrm{mod}\ n)$. Nếu n là số nguyên tố thì $x\equiv 1$ hoặc $x\equiv -1 (\mathrm{mod}\ n)$, ngược lại thì n phải là hợp số (bài tập 2.22).

Ví dụ, với số Carmicheal bé nhất 561 ta có: $5^{(561-1)/2}=5^{280} \equiv 67(\text{mod } 561)$. Vậy, 561 là hợp số.

Về sau, ta sẽ đề cập đến những thuật toán kiểm tra nguyên tố hiện đại. Trong phần này, để thấy thêm ứng dụng của các định lí đồng dư vừa trình bày, ta tìm hiểu vài thuật toán đơn giản.

Định nghĩa 2.18. Giả sử n là số nguyên dương lẻ, $n-1=2^s t$, trong đó s là số nguyên không âm, t là số nguyên dương lẻ. Ta nói *n trải qua được kiểm tra Miller cơ sở b*, nếu hoặc $b^t \equiv 1(\text{mod } n)$, hoặc $b^{2^j t} \equiv -1(\text{mod } n)$, với j nào đó, $0 \le j \le s-1$.

Ta chứng tỏ rằng, nếu n là số nguyên tố thì *n trải qua được kiểm tra Miller cơ sở b với mọi số b sao cho $n|b$*. Thật vậy, giả sử $n-1=2^s t$. Đặt $x_k=b^{(n-1)/2^k}=b^{2^{s-k_t}}$, với $k=0,1,...,s$. Vì n là số nguyên tố nên $x_0 \equiv 1(\text{mod } n)$. Do đó $x_1^2 \equiv 1(\text{mod } n)$, tức là $x_1 \equiv 1(\text{mod } n)$ hoặc $x_1 \equiv -1(\text{mod } n)$. Tiếp tục quá trình như vậy ta sẽ đi đến kết luận rằng, hoặc $x_k \equiv 1(\text{mod } n)$ với $k=0,1,...,s$, hoặc $x_k \equiv -1(\text{mod } n)$ với một số nguyên k nào đó. Như vậy n trải qua được kiểm tra Miller cơ sở b.

Dễ thấy rằng, nếu n trải qua được kiểm tra Miller cơ sở b thì n sẽ là số giả nguyên tố cơ sở b. Ta có định nghĩa sau.

Định nghĩa 2.19. n được gọi là *số giả nguyên tố mạnh cơ sở b* nếu nó là hợp số và trải qua được kiểm tra Miller cơ sở b.

Như vậy các số giả nguyên tố mạnh lại còn ít hơn các số giả nguyên tố. Tuy nhiên, ta có định lí sau.

Định lí 2.20. *Tồn tại vô số số giả nguyên tố mạnh cơ sở 2*.

Thật vậy, giả sử n là một số giả nguyên tố cơ sở 2. Khi đó, $2^{n-1}=nk$ với số nguyên lẻ k nào đó. Đặt $N=2^n-1$, ta có

$$N-1=2^n-2=2(2^{n-1}-1)=2nk;$$

nghĩa là n là hợp số. Mặt khác,

$$2^{(N-1)/2}=2^{nk}=(2^n)^k \equiv 1(\text{mod } N).$$

Vậy với mỗi số giả nguyên tố n, ta xây dựng được số giả nguyên tố mạnh N và các số n khác nhau cho ta các số N khác nhau: định lí được chứng minh, bởi vì có vô số giả nguyên tố cơ sở 2.

Ta có thể dùng kiểm tra Miller để kiểm tra nguyên tố những số không lớn lắm. Ta biết rằng, số giả nguyên tố mạnh lẻ có sở 2 bé nhất là 2047. Như vậy, nếu n lẻ và $n<2047$, thì n là nguyên tố nếu nó trải qua kiểm tra Miller. Tương tự như vậy, số 1373653, là số giả nguyên tố mạnh lẻ bé nhất cơ sở 2 và 3, được dùng để kiểm tra nguyên tố những số bé hơn nó. Đối với cơ sở 2,3 và 5, số giả nguyên tố mạnh lẻ bé nhất là 25326001, trong trường hợp cơ sở 2,3,5,7,số tương ứng là 3215031751. Trong những số nhỏ hơn 25.10^9, chỉ có một số giả nguyên tố lẻ với cơ sở 2,3,5,7, đó là 3251031751. Như vậy, nếu $n<25.10^9$ là số lẻ trải qua kiểm tra Miller, thì n là số nguyên tố nếu nó khác với 3251031751.

Cách làm trên đây chỉ áp dụng được khi cần kiểm tra nguyên tố những số không lớn. Đối với những số lớn, ta có thể dùng thuật toán xác suất dựa trên định lí sau đây:

Định lí 2.21. *Nếu n là một hợp số dương lẻ thì tồn tại không quá (n-1)/4 cơ sở b, $1 \le b \le n-1$, sao cho n trải qua được kiểm tra Miller đối với các cơ sở đó.*

Định lí trên đây được chứng minh dựa vào khái niệm chỉ số mà ta không trình bày ở đây. Độc giả nào quan tâm có thể tìm đọc trong[Ro]. Nhờ định lí 2.21, ta có thể kết luận *n* là một hợp số nếu thấy nó trải qua kiểm tra Miller với hơn (n-1)/4 cơ sở. Tuy nhiên, việc kiểm tra như thế đòi hỏi quá nhiều thời gian.

Từ định lí 2.21 suy ra rằng, nếu số *b* được chọn ngẫu nhiên trong khoảng $1 \le b \le n-1$ thì *n* trải qua kiểm tra Miller cơ sở *b* với xác suất bé hơn 1/4. Như vậy, nếu ta chọn *k* số ngẫu nhiên thì xác suất để *n* trải qua kiểm tra Miller đối với *k* cơ sở đó sẽ bé hơn $1/4^k$. Khi *k* đủlớn, ví dụ *k*=20, xác suất đó quá nhỏ, nên với *n* trải qua với 20 cơ sở ngẫu nhiên thì có thể tin "hầu chắc chắn" rằng *n* là số nguyên tố. Từ đó ta có thuật toán xác suất sau đây.

Thuật toán Rabin-Miller (1980)

Cho $N \ge 3$ lẻ, thuật toán sau đây xác định rằng *N* là một hợp số, hoặc in ra thông báo *N* là số nguyên tố với xác suất lớn hơn $1-1/4^{20}$.

RM1. (Xuất phát). Đặt $q \leftarrow N-1$, $t \leftarrow 0$, và nếu q chẵn đặt $q \leftarrow q/2$, $t \leftarrow t+1$ (bây giờ ta có $N-1=2^t q$, với q lẻ). Sau đó đặt $c \leftarrow 20$.

RM2. (Chọn a mới). Chọn ngẫu nhiên số a trong khoảng $1<a<N$. Đặt $e \leftarrow 0$, $b \leftarrow a^q \bmod N$. Nếu $b=1$, chuyển sang RM4.

RM3.(Bình phương). Nếu $b \not\equiv \pm 1 \pmod N$ và $e<t-2$, ta đặt $b \leftarrow b^2 \bmod N$, $e \leftarrow e+1$. Nếu $b \ne N-1$, in ra thông báo "n là hợp số" và kết thúc thuật toán.

RM4. Đặt $c \leftarrow c-1$. Nếu $c>0$, chuyển sang RM2. Nếu $c=0$, in ra thông báo "N là số nguyên tố".

§7. Phân số liên tục.

Giả sử *a,b* là các số nguyên dương, *a>b*. Khi đó, phân số *a*/*b* có thể viết dưới dạng:

$$\frac{a}{b} = a_0 + \frac{c_0}{b} = a_0 + \frac{1}{\dfrac{b}{c_0}}.$$

Phân số b/c_0 lại có thể biểu diễn dưới dạng tương tự như vậy, và cuối cùng ta nhận được:

$$\frac{a}{b} = a_o + \cfrac{1}{a_1 + \cfrac{1}{\ldots a_{n-1} + \cfrac{1}{a_n}}}.$$

Cách viết như trên được gọi là biểu diễn số hữu tỷ a/b dưới dạng *phân số liên tục*.

Để đơn giản kí hiệu, ta thường dùng cách viết $a/b=[a_0;a_1,a_2,...,a_n]$. Phân số liên tục $[a_0;a_1,a_2,...,a_n]$ được gọi là *phân số liên tục hữu hạn*.

Dùng thuật toán Euclid, có thể biểu diễn mọi số hữu tỷ dưới dạng phân số liên tục hữu hạn. Thật vậy, ta có $a=a_0b+c_0$, $b=a_1c_0+c_1$,.... Ngược lại, rõ ràng mỗi phân số hữu hạn liên tục là một số hữu tỷ.

Ta cũng có thể biểu diễn một số thực tuỳ ý dưới dạng phân số liên tục. Tuy nhiên trong trường hợp này, phân số liên tục có thể không hữu hạn. Cách làm cũng hoàn toàn tương tự như khi làm với các số hữu tỷ.

Giả sử x là số thực tuỳ ý. Đặt $a_0=[x]$, phần nguyên của x, và $x_0=x-a_0$ là phần lẻ của x. Tiếp theo đó, ta đặt $a_1=[1/x_0]$, $x_1=1/x_0-a_1$. Tóm lại đối với mỗi số $i>1$, đặt $a_i=[1/x_{i-1}]$, $x_i=1/x_{i-1}-a_i$. Nếu ở bước thứ i nào đó, $x_i=0$ thì quá trình kết thúc (Điều này xảy ra khi và chỉ khi x là số hữu tỷ). Ngược lại, ta có biểu diễn x dưới dạng phân số liên tục vô hạn: $[a_0;a_1,a_2,...,a_n,...]$.

Nhiều khi để thuận tiện, ta dùng cách viết sau đây:

$$x= a_0 + \frac{1}{a_1 +} + \frac{1}{a_2 +} + \ldots + \frac{1}{a_n +} + \ldots$$

Các phân số liên tục định nghĩa như trên với các số a_i nguyên còn được gọi là các *phân số liên tục đơn giản*. Khi không đòi hỏi a_i là các số nguyên, mà có thể là các số thực tuỳ ý, ta cũng dùng cách viết

$$x=[a_0;a_1,a_2,...,a_n]= \quad a_0 + \frac{1}{a_1 +} + \frac{1}{a_2 +} + \ldots + \frac{1}{a_n}.$$

Khi có một phân số liên tục $x=[a_0;a_1,a_2,...,a_n,...]$, ta gọi các số sau đây là các *phân số hội tụ riêng của x:*

$$C_k= [a_0;a_1,a_2,...,a_k].$$

Định lí 2.22. *Giả sử $a_0,a_1,...,a_n$ là các số thực, trong đó $a_0,a_1,...,a_n>0$. Đặt $p_0=a_0$, $q_0=1$, $p_1=a_0a_1+1$, $q_1=a_1$, và với mỗi $k\geq 2$, $p_k=a_kp_{k-1}+p_{k-2}$, $q_k=a_kq_{k-1}+q_{k-2}$. Khi đó đối với các phân số hội tụ riêng C_k ta có:*

$$C_k= [a_0;a_1,a_2,...,a_k]= p_k/q_k.$$

Chứng minh. Ta chứng minh bằng qui nạp. Với $k=0$, $C_0=a_0=p_0/q_0$. Với $k=1$,

$$C_1=[a_0;a_1]=a_0+\frac{1}{a_1}=p_1/q_1.$$

Ta có:

$$C_{k+1}=[a_0;a_1,a_2,...,a_{k+1}]=a_0+\frac{1}{a_1+}+\frac{1}{a_2+}+\ldots+\frac{1}{a_{k+1}}$$

$$=[a_0;a_1,a_2,...,a_{k-1},a_k+\frac{1}{a_{k+1}}]=\frac{(a_k+\frac{1}{a_{k+1}})p_{k-1}+p_{k-2}}{(a_k+\frac{1}{a_{k+1}})q_{k-1}+q_{k-2}}.$$

theo giả thiết qui nạp. Tính toán đơn giản dựa vào định nghĩa các số p_k,q_k, ta được:

$$C_{k+1}=p_{k+1}/q_{k+1.}$$

Định lí được chứng minh.

Định lí 2.23. *Với mọi* $k\geq 1$, *ta có:*

$$p_kq_{k+1}-p_{k-1}q_k=(-1)^{k-1}.$$

Từ đó ta suy ra ngay rằng, các số p_k,q_k nguyên tố cùng nhau.

Định lí 2.24. *Ta có:*

$$C_1>C_3>C_5>...$$

$$c_0<C_2<C_4>...$$

$$C_{2j+1}>C_{2k}, \text{ với mọi } j,k$$

$$\lim C_k=x.$$

Chứng minh các định lí trên (bằng quy nạp) được dành cho độc giả. Có thể thấy rằng, tên gọi "phân số liên tục riêng " được giải thích bằng định lí trên đây.

Định lí 2.25. *Giả sử n là một số tự nhiên không chính phương và* p_k,q_k *là các phân số hội tụ riêng của* $\sqrt{n}$. *Ta đặt* $\alpha_0=\sqrt{n}$, *và các số* α_k, Q_k, P_k *được định nghĩa theo công thức sau:*

$$\alpha_k=(P_k+\sqrt{n})/Q_k,$$

$$a_k=[\alpha_k],$$

$$P_{k+1}=a_kQ_k-P_k$$

$$Q_{k+1}=(n-P_{n+1}^2)Q_k$$

Khi đó ta có:

$$p_k^2-n\,q_k^2=(-1)^{k-1}Q_{k+1}.$$

Chứng minh. Áp dụng định lí vừa chứng minh, ta có:

$$\sqrt{n} = \alpha_0 = [a_0; a_1, a_2, \ldots, a_{k+1}] = \frac{a_{k+1}p_k + p_{k-1}}{a_{k+1}q_k + q_{k-1}}.$$

Từ đó, vì $a_{k+1} = (P_{k+1} + \sqrt{n})/ Q_{k+1}$, ta được:

$$\sqrt{n} = \frac{(P_{k+1} + \sqrt{n})p_k + Q_{k+1}p_{k-1}}{(P_{k+1} + \sqrt{n})q_k + Q_{k+1}q_{k-1}}.$$

Vậy,

$$nq_k + (P_{k+1}q_k + Q_{k+1}q_{k-1})\sqrt{n} = (P_{k+1}p_k + Q_{k+1}p_{k-1}) + p_k\sqrt{n}.$$

Từ đó suy ra:

$$nq_k = P_{k-1}p_k + Q_{k+1}p_{k-1},$$

$$p_k = P_{k+1}q_k + Q_{k+1}q_{k-1}.$$

Nhân đẳng thức thứ nhất với q_k, đẳng thức thứ hai với p_k và trừ đẳng thức thứ hai cho đẳng thức thứ nhất, ta thu được kết quả cần chứng minh.

Sau đây ta sẽ áp dụng phân số liên tục để tìm một thuật toán phân tích số nguyên ra thừa số nguyên tố. Nói chính xác hơn, ta sẽ xây dựng một thuật toán để với số tự nhiên n cho trước, tìm ước số không tầm thường (khác 1 và n).

Ta xuất phát từ nhận xét đơn giản sau đây: Nếu ta tìm được các số x, y sao cho $x-y \neq 1$ và $x^2-y^2=n$ thì ta tìm được số ước không tầm thường của n, vì $n = x^2-y^2=(x-y)(x+y)$.

Bây giờ, giả sử ta có kết quả yếu hơn, chẳng hạn tìm được x,y sao cho $x^2 \equiv y^2 \pmod n$ và $0<x<y<n$, $x+y \neq n$.

Khi đó n là một ước của tích $(x-y)(x+y)$, và rõ ràng n không là ước của $x+y$ cũng như $x-y$. Như vậy các ước số chung $d_1=(n,x-y)$ và $d_2=(n,x+y)$ là các ước số không tầm thường của n. Các ước số này tìm được một cách nhanh chóng nhờ thuật toán Euclid. Định lí 2.25 cho ta phương pháp để tìm các số x,y cần thiết.

Theo định lí 2.25 ta có:

$$p_k^2 \equiv (-1)^{k-1}Q_{k-1} \pmod n.$$

Như vậy, vấn đề là phải tìm được các Q_k với chỉ số chẵn, và là một số chính phương. Mỗi lần tìm được một số Q_k như vậy, ta tìm được một ước của n (cũng có thể xảy ra trường hợp ước đó là tầm thường: các p_k, Q_k không nhất thiết bé hơn n nên điều kiện n không phải là ước của x+y và x-y có thể không thoả mãn).

Ví dụ. 1). Phân tích số 1037 ra thừa số bằng cách sử dụng phân số liên tục.

ta có $\alpha = \sqrt{1037} = 32,2...$, $Q_1=1, Q_2=49$, $p_1=129$. Như vậy, $129^2 \equiv 49 \pmod{1037}$. Do đó, $129^2-7^2=(129-7)(129+7) \equiv 0 \pmod{1037}$. Tính các ước chung lớn nhất, ta được: $(129-7,1037)=61$, $(129+7,1037)=17$. Ta có hai ước của 1037, và trong trường hợp này có khai triển $1037=61.17$.

2) Phân tích 1000009. Ta tính được $Q_1=9$, $Q_2=445$, $Q_3=873$, $Q_4=81$. Như vậy, $p_3^2 \equiv 9^2 (\mod 100009)$: ta không thu được ước không tầm thường. Tính toán tiếp tục, ta có: $Q_{18}=16$ là một số chính phương, $p_{17}=494881$. Bằng thuật toán đã mô tả, ta tìm được các ước số 293, 3413.

Bài tập và tính toán thực hành chương 2

I. Bài tập.

2.1. Chuyển số (1999) từ cơ số 10 sang cơ số 7, số (6105) từ cơ số 7 sang cơ số 10.

2.2. Chuyển các số 10001110101 và 11101001110 từ cơ số 2 sang cơ số 16 (kí hiệu các chữ số của cơ số 16 bởi 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E).

2.3. Chứng minh rằng mọi vật nặng không quá 2^k-1 (với trọng lượng là số nguyên) đều có thể cân bằng một cái cân hai đĩa, sử dụng các quả cân 1, 2, 2^2,...,2^{k-1}.

2.4. Chứng minh rằng mọi số nguyên đều có thể biểu diễn duy nhất dưới dạng

$$e_k 3^k + e_{k-1} 3^{k-1} + ... + e_1 3 + e_0,$$

trong đó e_j=-1, 0 hoặc 1; j=0,1,...,k.

2.5. Chứng minh rằng, mọi số thực $\alpha \in R$, $0 \le \alpha < 1$ đều có thể biểu diễn duy nhất dưới dạng cơ số *b*

$$\alpha = \sum_{j=1}^{\infty} c_j / b^j \ , \ 0 \le c_j \le b\text{-}1,$$

thoả mãn điều kiện: với mọi N, tồn tại $n \ge N$ để $c_n \ne b$-1.

2.6. Áp dụng bài 2.5, viết π trong cơ số 2 với 10 chữ số sau dấu phẩy.

2.7. a) Chứng minh rằng mọi số nguyên dương *n* đều có biểu diễn Cantor duy nhất dưới dạng sau:

$$n = a_m m! + a_{m-1}(m-1)! + ... + a_2 2! + a_1 1!.$$

b) Tìm khai triển Cantor của 14, 56, 384.

2.8. Giả sử *a* là số nguyên (trong cơ số 10) với bốn chữ số sao cho không phải mọi chữ số là như nhau. *a*' là số nhận được từ *a* bằng cách viết các chữ số theo thứ tự giảm dần, *a*'' là số nhận được bằng cách viết các chữ số theo thứ tự tăng dần. Đặt T(a)=a'-a''. Ví dụ: T(1998)=9981-1899.

a) Chứng minh rằng số nguyên duy nhất (không phải 4 chữ số đều như nhau) sao cho T(a)=a là a=6174.

b) Chứng minh rằng nếu *a* là số nguyên dương 4 chữ số, không phải mọi chữ số đều như nhau, thì dãy *a, T(a), T(T(a)),...* nhận được bằng cách lặp phép toán T, sẽ dừng ở số 6174 (được gọi là *hằng số Kapreka*)

2.9. Ước lượng thời gian cần thiết để tính *n*!.

2.10. Ước lượng thời gian cần thiết để chuyển một số *k*-bit sang hệ thập phân.

2.11. a) Chứng minh rằng, nếu A, B là các ma trận vuông cấp *n* thì để tìm tích AB (theo quy tắc nhân ma trận thông thường) ta cần n^3 phép nhân.

b) Chứng minh rằng có thể nhân hai ma trận vuông cấp hai mà chỉ cần 7 phép nhân, nếu sử dụng đồng nhất thức sau đây:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}=$$

$$\begin{pmatrix} a_{11}\ b_{11} + a_{12}b_{21} & x + (a_{21} + a_{22})(b_{12} - b_{11}) + (a_{11} + a_{12} - a_{21} - a_{22})b_{22} \\ x + (a_{11} - a_{21})(b_{22} - b_{12}) - a_{22}(b_{11} - b_{21} - b_{12} + b_{22}) & x + (a_{11} - a_{21})(b_{22} - b_{12}) + (a_{21} + a_{22})(b_{12} - b_{11}) \end{pmatrix},$$

trong đó $x=a_{11}b_{11}-(a_{11}-a_{21}-a_{22})(b_{11}-b_{12}+b_{22})$.

c) Bằng quy nạp và tách ma trận $2n \times 2n$ thành 4 ma trận $n \times n$, chứng minh rằng có thể nhân hai ma trận $2^k \times 2^k$ chỉ với 7^k phép nhân và không ít hơn 7^{k+1} phép cộng.

d) Từ c) suy ra rằng có thể nhân hai ma trận vuông cấp n với $O(n^{\log}c^7)$ phép tính bit nếu mọi phần tử của ma trận có dưới c bit, với hằng số c nào đó.

2.12. Dùng sàng Eratosthenes để tìm mọi số nguyên tố bé hơn 1998.

2.13. Cho $Q_n=p_1p_2...p_n+1$, trong đó $p_1,p_2,...,$ p_n là n số nguyên tố đầu tiên. Tìm ước nguyên tố bé nhất của Q_n, với n=1,2,3,4,5,6.

Trong dãy Q_n có vô hạn hay hữu hạn số nguyên tố?

2.14. Chứng minh rằng tồn tại vô hạn số nguyên tố.

2.15. Chứng minh rằng nếu ước nguyên tố bé nhất p của một số nguyên dương n vượt quá $\sqrt[3]{n}$ thì n/p là số nguyên tố.

2.16. Chứng minh rằng không tồn tại một "bộ ba nguyên tố" nào p, p+2, p+4 ngoài 3,5,7.

2.17. Chứng minh rằng nếu a|x, b|x và a, b nguyên tố cùng nhau thì a.b|x.

2.18. Chứng minh rằng nếu a,m nguyên tố cùng nhau thì tồn tại nghịch đảo m mod b.

2.19. Cho a,b,c,m là các số nguyên, m dương. Giả sử d là ƯCLN của c và m. Khi đó, nếu $ac \equiv bc(\text{mod } m)$ thì $a \equiv b(\text{mod } m/d)$.

2.20. Giả sử $r_1, r_2,...,r_m$ là một hệ thặng dư đầy đủ modulo m, a là một số nguyên, nguyên tố cùng nhau với m, b là số nguyên tuỳ ý. Chứng minh rằng $ar_1+b, ar_2+b ,..., ar_m+b$ cũng là một hệ đầy đủ các thặng dư modulo m

2.21. Giả sử $a \equiv b(\text{mod } m_j)$, j=1,2,..., k, trong đó m_j là các số nguyên tố cùng nhau từng cặp. Chứng minh rằng $a \equiv b(\text{mod } m_1m_2...m_k)$.

2.22. Cho p là số nguyên tố. Chứng minh rằng $a^2 \equiv 1(\text{mod } p)$ khi và chỉ khi $a \equiv \pm 1(\text{mod } p)$.

2.23. Chứng minh rằng với mọi số nguyên không âm m,n và mọi số nguyên $a>1$, ta có

$$(a^m-1, a^n-1)=a^{(m,n)}-1.$$

2.24. a) Chứng minh rằng có thể tìm ước chung lớn nhất của hai số nguyên dương bằng thuật toán sau

$$(a,b)=\begin{cases} a & \text{nếu } a=b \\ 2(a/2,b/2) & \text{nếu } a,b \text{ chẵn} \\ (a/2,b) & \text{nếu } a \text{ chẵn}, b \text{ lẻ} \\ (a-b,b) & \text{nếu } a,b \text{ lẻ} \end{cases}$$

b) Dùng thuật toán trên để tìm (2106, 8318).

2.25. Chứng minh rằng, với mọi n, tìm được n số tự nhiên liên tiếp sao cho mỗi số đều có ước là số chính phương.

2.26. Giả sử $n=p_1p_2...p_k$, trong đó p_j là các số nguyên tố, và n là một số Carmichael. Chứng minh rằng $k\geq 3$. Áp dụng kết quả để tìm ra số Carmichael nhỏ nhất.

2.27. Chứng minh rằng, nếu 6m+1, 12m+1, 18m+1 đều là số nguyên tố thì (6m+1)(12m+1)(18m+1) là số Carmichael.

Chứng minh các số sau đây là số Carmichael:

1729, 294409, 56052361, 118901521, 172947529.

2.28. Chứng minh rằng 6601 là một số Carmichael.

2.29. Chứng minh rằng n=2047=23.89 là số giả nguyên tố mạnh cơ sở 2.

2.30. Cho b,m là các số nguyên nguyên tố cùng nhau, a,c là các số nguyên dương. Chứng minh rằng, nếu $b^a\equiv 1(\text{mod } m)$, $b^c\equiv 1(\text{mod } m)$ và $d=(a,c)$ thì $b^d\equiv 1(\text{mod } m)$.

2.31. Cho p là số nguyên tố, $p|b^m-1$. Chứng minh rằng, hoặc $p|b^d-1$ với d nào đó là ước thực sự của m (khác m), hoặc $d\equiv 1(\text{mod } m)$. Nếu $p>2$, m lẻ thì trong trường hợp sau, ta có $p\equiv 1(\text{mod } 2n)$.

2.32. Áp dụng bài tập trên để phân tích ra thừa số các số $2^{11}-1=2047$, $2^{13}-1=8191$, $3^{12}-1=531440$, $2^{35}-1=34355738367$.

2.33. Tìm phân số liên tục của các số $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $(1+\sqrt{5})/2$.

2.34. Biết phân số liên tục của e là

e=[2;1,2,1,1,4,1,1,6,1,1,8,...]

a) Tìm 8 phân số hội tụ riêng đầu tiên của e.

b) Tìm xấp xỉ hữu tỷ tốt nhất của e có mẫu số bé hơn 100.

2.35. Cho α là một số vô tỷ. Chứng minh rằng, hoặc $|\alpha-p_k/q_k|<1/2q_k^2$, hoặc $|\alpha-p_{k+1}/q_{k+1}|<1/2q_{k+1}^2$.

2.36. Cho $f(x)$ là một đa thức tuỳ ý với hệ số nguyên. Chứng minh rằng tồn tại vô hạn số nguyên k sao cho $f(k)$ là hợp số.

II. Thực hành tính toán trên máy

Đối với tất cả các chương, tính toán thực hành trên máy tính với chương trình Maple được bắt đầu bằng dòng lệnh:

```
[>with(numtheory);
```

Các phép toán số học (phép cộng [+], phép trừ [-], phép nhân [*], phép chia [/], phép luỹ thừa [^], khai căn bậc hai [sqrt(.)],...) được viết và thực hiện theo thứ tự quen biết.

Luôn luôn ghi nhớ rằng cuối dòng lệnh phải là dấu chấm phẩy (;) hoặc dấu (:). Muốn thực hiện dòng lệnh nào thì phải đưa con trỏ về dòng lệnh đó (sau dấu chấm phẩy) và nhấn phím [Enter]. Hãy thực hiện các dòng lệnh theo đúng trình tự trước sau, vì một số tính toán trong các bước sau có thể yêu cầu kết quả từ các bước trước.

II. 1. Thực hành kiểm tra một số là số nguyên tố

Để kiểm tra một số *n* có phải là số nguyên tố hay không ta thực hiện lệnh như sau:

```
[>isprime(n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện ra chữ "true" thì *n* là số nguyên tố, nếu trên màn hình hiện ra chữ "false" thì *n* là hợp số.

Thí dụ: Số 2546789 có phải là số nguyên tố hay không?

```
[>isprime(n);

                              False
```

Vậy 2546789 không phải là số nguyên tố.

II. 2. Thực hành tìm ước chung lớn nhất

Để thực hành tìm ước chung lớn nhất của hai số a và b, hãy vào dòng lệnh có cú pháp như sau:

```
[>gcd(a,b);
```

Sau dấu (;) ấn phím "Enter" thì việc tìm ước chung lớn nhất sẽ được thực hiện và sẽ có ngay kết quả.

Thí dụ: Tìm ước số chung lớn nhất của 2 số 157940 và 78864.

Thực hiện bằng câu lệnh sau:

```
[> gcd(157940,78800);

                               20
```

Vậy ước chung lớn nhất của 157940 và 78864 là 20.

II. 3. Phân tích ra thừa số nguyên tố

Để phân tích số *n* ra thừa số nguyên tố ta thực hiện lệnh sau:

```
[>ifactor(n);
```

Sau dấu (;) ấn phím "Enter" thì việc phân tích *n* ra thừa số nguyên tố sẽ được thực hiện và sẽ có ngay kết quả.

Thí dụ: Phân tích số 122333444455555666666777777788888888999999999 ra thừa số nguyên tố.

Ta thực hiện như sau:

```
[>
ifactor(122333444455555666666777777788888888999999999);

   (3)(12241913785205210313897506033112067347143)(3331)
```

Ta cũng có thể dùng lệnh trên để kiểm tra xem một số *n* có phải là số nguyên tố hay không

II. 4. Thực hành kiểm tra một số là số Carmichael

Ta nhớ lại Định lí 2. 17 như sau:

Định lí 2.17. *Nếu* $n=q_1q_2...q_k$, *trong đó* q_j *là các số nguyên tố khác nhau thoả mãn* $(q_j-1) | (n-1)$, *thì n là số Carmichael*.

Do đó để kiểm tra xem một số *n* có phải là số Carmichael hay không ta thực hiện theo các bước sau:

Bước 1: Phân tích *n* thành tích các thừa số nguyên tố, ta thực hiện bằng dòng lệnh:

```
[>ifactor(n);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả phân tích *n* ra thừa số nguyên tố. Nếu *n* là hợp số và có dạng $n=q_1q_2...q_k$, trong đó q_j là các số nguyên tố khác nhau thì thực hiện tiếp bước kiểm tra thứ 2. Nếu không thì có thể khẳng định *n* không phải là số Carmichael.

Bước 2:. Thực hiện các phép tính chia $(n-1):(q_j-1)$, ta thực hiện bằng dòng lệnh sau:

```
[>(n-1)/(q_j-1);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả thương của phép chia**.** Nếu với mọi $j=1,2, ..., k$ các thương tìm được là các số nguyên thì ta khẳng định *n* là số Carmichael, nếu không thì trả lời không phải.

Thí dụ 1: Số 6601 có phải là số Carmichael hay không?

Thực hiện kiểm tra như sau:

```
[>ifactor(6601);
                          (7)(23)(41)
```

6601 được phân tích thành các thừa số nguyên tố khác nhau, vậy có thể nghi ngờ nó là số Carmichel. Để kiểm tra xem nó có thực sự là số Carmichel hay không, ta thực hiện các lệnh sau:

```
[>(6601-1)/(7-1);
                              1100
[>(6601-1)/(23-1);
                               300
[>(6601-1)/(41-1);
                               165
```

Vậy 6601 là số Carmichael.

Thí dụ 2: Số 6 có phải là số Carmichael hay không?

Thực hiện kiểm tra như sau:

```
[>ifactor(6);
                             (2)(3)
[>(6-1)/(2-1);
                                5
[>(6-1)/(3-1);
                               5
                               2
```

Vậy 6 không phải là số Carmichael.

Thí dụ 3: Số 45 có phải là số Carmichael hay không?

Thực hiện kiểm tra như sau:

```
[>ifactor(45);
                           (3)^2(5)
```

Số 45 không thoả mãn bước thứ nhất.

Vậy 45 không phải là số Carmichael.

II. 5. Thực hành kiểm tra một số là giả nguyên tố

Cho hai số nguyên dương *n, b*. Để kiểm tra xem *n* có phải là số giả nguyên tố cơ sở *b* hay không ta thực hiện các bước như sau:

Bước 1: Kiểm tra *n* là hợp số, ta thực hiện dòng lệnh:

```
[>isprime(n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện ra chữ "true" thì *n* là số nguyên tố, nếu trên màn hình hiện ra chữ "false" thì *n* là hợp số. Nếu *n* là số nguyên tố thì *n* không phải là số giả nguyên tố cơ sở *b*. Nếu ngược lại thực hiện tiếp bước 2.

Bước 2: Kiểm tra đồng dư thức $b^n-b \equiv 0(mod\ n)$, thực hiện bằng dòng lệnh:

```
[>b&^n-b mod n;
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả. Nếu đó là số 0 thì *n* là số giả nguyên tố cơ sở *b*.

Thí dụ1: Số 561 có phải là số giả nguyên tố cơ sở 2 hay không?

Ta thực hiện các lệnh sau:

```
[>isprime(561);
                              false
[>2&^561-2 mod 561;
                                0
```

Vậy 561 là số giả nguyên tố cơ sở 2.

Thí dụ 2: Số 1224191378520521031389750603311206734714 3 có phải là số giả nguyên tố cơ sở 8 hay không?

Ta thực hiện các lệnh sau:

```
[>ispime(1224191378520521031389750603311206734714 3);
                              true
```

Số 1224191378520521031389750603311206734714 3 là một số nguyên tố. Do đó 1224191378520521031389750603311206734714 3 không phải là số giả nguyên tố cơ sở 8.

Thí dụ 3: Số 326 có phải là số giả nguyên tố cơ sở 3 hay không?

Ta thực hiện các lệnh sau:

```
[>isprime(326);
                              false
[>3&^326-3 mod 326;
                                6
```

Vậy 326 là không phải là số giả nguyên tố cơ sở 3.

II. 6. Thực hành kiểm tra một số là số giả nguyên tố mạnh

Cho *n* là số nguyên dương lẻ, *b* là số nguyên dương. Để kiểm tra *n* có phải là số giả nguyên tố mạnh cơ sở *b* hay không ta thực hiện theo các bước sau:

Bước 1: Kiểm tra *n* là hợp số, ta thực hiện bằng dòng lệnh:

```
[>isprime(n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện ra chữ "true" thì n là số nguyên tố, nếu trên màn hình hiện ra chữ "false" thì n là hợp số. Nếu n là số nguyên tố thì n không phải là số giả nguyên tố mạnh cơ sở b. Nếu ngược lại thực hiện tiếp bước 2.

Bước 2: Phân tích n-1 ra thừa số nguyên tố, ta thực hiện bằng dòng lệnh:

```
[>ifactor(n-1);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra sự phân tích của n-1 và ta thu được kết quả có dạng $n-1=2^s t$, trong đó s là số nguyên dương, t là số nguyên dương lẻ.

Bước 3: Kiểm tra đồng dư thức $b^t-1 \equiv 0 (mod\ n)$. Vào lệnh

```
[>b&^t-1 mod n;
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả. Nếu đó là số 0 thì n là số giả nguyên tố mạnh cơ sở b, nếu kết quả là một số khác 0 ta thực hiện tiếp bước 4.

Bước 4: Kiểm tra các đồng dư thức $(b^{2^j t}+1) \equiv 0 (mod\ n)$ với j=0,...s-1, ta thực hiện dòng lệnh:

```
[>seq (b&^((2^j)t)+1 mod n, j=0..s-1);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện ra dãy kết quả. Nếu trong dãy kết quả có một số là số 0 thì n là số giả nguyên tố mạnh cơ sở b.

Thí dụ: Số 2047 có phải là số giả nguyên tố mạnh cơ sở 2 hay không?

Thực hiện kiểm tra như sau:

```
[>isprime(2047);
                              false
```

Do đó n là hợp số. Tiếp tục thực hiện lệnh

```
[>ifactor(n-1);
                          (2)(3)(11)(31)
```

Tiếp tục thực hiện lệnh

```
[>2&^(3*11*31)-1 mod 2047;
                                0
```

Vậy 2047 là số giả nguyên tố mạnh cơ sở 2.

II. 7. Thực hành biểu diễn một số dưới dạng phân số liên tục

1. Biểu diễn số n dưới dạng phân số liên tục theo cách thông thường với số thương trong biểu diễn là k, ta dùng lệnh:

```
[>cfrac(n,k);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Biểu diễn π dưới dạng phân số liên tục theo cách thông thường với 6 thương.

Ta thực hiện lệnh:

```
[> cfrac (Pi,6);
```

$$3+\cfrac{1}{7+\cfrac{1}{15+\cfrac{1}{1+\cfrac{1}{292+\cfrac{1}{1+\cfrac{1}{1+\ldots}}}}}}$$

2. Biểu diễn số *n* dưới dạng phân số liên tục theo cách đơn giản với số chữ số trong biểu diễn là *k*, ta dùng lệnh:

```
[>cfrac(n,k,'quotients');
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Biểu diễn π dưới dạng phân số liên tục theo cách viết đơn giản với 100 chữ số biểu diễn.

Ta thực hiện lệnh:

```
[> cfrac (Pi,100,'quotients');

[3,7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1,84,2,1,1,
15,3,13,1,4,2,6,6,99,1,2,2,6,3,5,1,1,6,8,1,7,1,2,3,7,1,
2,1,1,12,1,1,1,3,1,1,8,1,1,2,1,6,1,1,5,2,2,3,1,2,4,4,16,
1,161,45,1,22,1,2,2,1,4,1,2,24,1,2,1,3,1,2,1,1,10,2,...]
```

3. Biểu diễn số *n* dưới dạng phân số liên tục theo chu kỳ tuần hoàn, ta dùng lệnh:

```
[>cfrac(n,'periodic');
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Biểu diễn $3^{1/2}$ dưới dạng phân số liên tục theo chu kỳ tuần hoàn.

Ta thực hiện lệnh:

```
[>cfrac (3^(1/2),'periodic');
```

$$1+\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{1+\cfrac{1}{2+\ldots}}}}$$

4. Biểu diễn số *n* dưới dạng phân số liên tục theo chu kỳ tuần hoàn đơn giản, ta dùng lệnh:

```
[>cfrac (n,'periodic','quotients');
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Biểu diễn $3^{1/2}$ dưới dạng phân số liên tục theo chu kỳ tuần hoàn đơn giản.

Ta thực hiện lệnh:

```
[> cfrac (3^(1/2),'periodic','quotients');
                                [[1], [1, 2]]
```

II. 8. Thực hành tìm phân số hội tụ thứ *k* của một số

Để thực hành tìm phân số hội tụ thứ *k* của một số *n*, ta thực hiện theo các lệnh sau:

Bước 1: Biểu diễn *n* dưới dạng phân số liên tục

```
[> cf:= cfrac(n);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện sự biểu diễn

Bước 2: Tính phân số hội tụ thứ *k*

```
[> nthconver(cf,k);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện ra kết quả.

Trong quá trình thực hiện ta không cần biết kết quả hiện thị ở bước 1, do đó có thể thay dấu (;) bằng dấu (:) ở dòng lệnh đầu tiên (`[>cf:=cfrac(n):`). Khi đó trên màn hình sẽ hiện ra dấu nhắc ([>) để thực hiện tiếp lệnh thứ 2.

Thí dụ: Tính phân số hội tụ thứ 5 của *e*.

Ta thực hiện như sau:

```
[> cf:= cfrac(exp(1));
```

$$cf := 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{6 + \cfrac{1}{1+\ldots}}}}}}}}}$$

```
[> nthconver(cf,5);
```

$$\frac{87}{32}$$

Như vậy, phân số hội tụ thứ 5 của e là $\frac{87}{32}$.

II. 8. Thực hành đổi cơ số

1. Để thực hành đổi một số n từ cơ 10 sang cơ số b ta dùng dòng lệnh sau:

```
[>convert(n,base,b);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên một dòng kết quả. Chú ý rằng kết quả đưa ra trên màn hình được viết theo thứ tự ngược lại.

Thí dụ 1: Đổi số 24564 từ cơ số 10 sang cơ số 6.

Ta thực hành như sau:

```
[>convert(24564,base,6);

                          [0, 2, 4, 5, 0, 3]
```

Vậy ta được số là $(305420)_6$.

Chú ý: Trong trường hợp cơ số b >10, ta vẫn thực hiện dòng lệnh đổi cơ số như bình thường. Tuy nhiên, sau khi nhận được kết quả, để tránh nhầm lẫn ta thực hiện việc đặt tương ứng các số lớn hơn 10 với các kí hiệu nào đó. Ta xem ví dụ sau:

Thí dụ 2: Đổi số 45676 từ cơ số 10 sang cơ số 15, trong đó đặt 10=A, 11=B,12=C,13=D,14=E.

Ta thực hành như sau:

```
[>L:=convert(45676,base,6):
[>subs(10=A,11=B,12=C,13=D,14=E,L);

                             [1, 0, 8, D]
```

Vậy ta được số là $(D801)_{15}$.

2. Để thực hành đổi một số n từ cơ số a sang cơ số b ta dùng dòng lệnh sau:

```
[> convert(n,base,a,b);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên một dòng kết quả. Chú ý rằng kết quả đưa ra trên màn hình được viết theo thứ tự ngược lại.

Thí dụ: Đổi số 305420 trong cơ số 6 sang cơ số 10.

Ta thực hiện dòng lệnh

```
[> convert([0,2,4,5,0,3],base,6,10);

                                [4, 6, 5, 4, 2]
```

Vậy ta có kết quả là $(24564)_{10}$

Chương 3

CÁC HÀM SỐ HỌC

Khi nghiên cứu các số nguyên, ta thường làm việc với các đại lượng như: số các ước của một số nguyên tố cho trước, tổng các ước của nó, tổng các luỹ thừa bậc k của các ước,... Ngoài những ví dụ đó còn có rất nhiều hàm số học quan trọng khác. Trong chương này, ta chỉ xét sơ qua một vài hàm quan trọng. Phần lớn của chương được giành cho hàm Euler, là một trong những hàm số học quan trọng nhất.

§1. Định nghĩa.

Định nghĩa 3.1. *Hàm số học* tức là hàm xác định trên tập hợp các số nguyên dương.

Định nghĩa 3.2. Một hàm số học f được gọi là *nhân tính* nếu với mọi n, m nguyên tố cùng nhau, ta có $f(mn)=f(m)f(n)$. Trong trường hợp đẳng thức đúng với mọi m,n (không nhất thiết nguyên tố cùng nhau), hàm f được gọi là *nhân tính mạnh*.

Những ví dụ đơn giản nhất về hàm nhân tính (mạnh) là: $f(n)=n$ và $f(n)=1$.

Dễ chứng minh tính chất sau đây: nếu f là một hàm nhân tính, n là số nguyên dương có khai triển thành thừa số nguyên tố dạng $n=p_1^{a1}p_2^{a2}...p_k^{ak}$, thì $f(n)$ được tính theo công thức

$$f(n)=f(p^{a1})f(p^{a2})..f(p^{ak}).$$

§2. Phi hàm Euler.

Trong các hàm số học, hàm Euler mà ta định nghĩa sau đây có vai trò rất quan trọng.

Định nghĩa 3.3. *Phi- hàm Euler* $\phi(n)$ là hàm số học có giá trị tại n bằng số các số không vượt quá n và nguyên tố cùng nhau với n.

Ví dụ. Từ định nghĩa ta có: $\phi(1)=1$, $\phi(2)=1$, $\phi(3)=2$, $\phi(4)=2$, $\phi(5)=4$, $\phi(6)=2$, $\phi(7)=6$, $\phi(8)=4$, $\phi(9)=6$, $\phi(10)=4$.

Từ định nghĩa trên đây ta có ngay hệ quả trực tiếp: Số p là nguyên tố khi và chỉ khi $\phi(p)=p-1$.

Nếu định lí Fermat bé cho ta công cụ nghiên cứu đồng dư modulo một số nguyên tố, thì Phi-hàm Euler được dùng để xét đồng dư modulo một hợp số. Trước khi đi vào vấn đề đó, ta cần một số định nghĩa sau.

Định nghĩa 3.4. *Hệ thặng dư thu gọn modulo n* là tập hợp ϕ(n) số nguyên sao cho mỗi phần tử của tập hợp nguyên tố cùng nhau với n , và không có hai phần tử nào đồng dư với nhau modulo n.

Nói cách khác từ hệ thặng dư đầy đủ modolo n, để lập hệ thặng dư thu gọn, ta chỉ giữ lại những giá trị nào nguyên tố cùng nhau với n.

Ví dụ. Các số 1,2,3,4,5,6 lập thành hệ thặng dư thu gọn modulo 7. Đối với modulo 8, ta có thể lấy 1,3,5,7.

Định lí 3.5. *Nếu* $r_1,r_2,...,r_{\phi(n)}$ *là một hệ thặng dư thu gọn modulo n, và a là số nguyên dương, (a,n)=1, thì tập hợp* $ar_1,ar_2,...,ar_{\phi(n)}$ *cũng là hệ thặng dư thu gọn modulo n.*

Chúng tôi dành chứng minh định lí này cho độc giả.

Định lí trên đây được dùng để chứng minh mở rộng của định lí Fermat bé.

Định lí Euler. *Nếu m là số nguyên dương và a là số nguyên tố cùng nhau với n thì* $a^{\phi(m)} \equiv 1(\text{mod } m)$.

Chứng minh. Ta lập luận hoàn toàn tương tự như trong định lí Fermat bé. Giả sử $r_1,r_2,...,r_{\phi(m)}$ modulo m, lập nên từ các số nguyên dương không vượt quá m và nguyên tố cùng nhau với m. Theo định lí 3.5, $ar_1,ar_2,...,a\,r_{\phi(m)}$ cũng là một hệ thặng dư thu gọn. Khi đó thặng dư dương bé nhất của hệ này sẽ là tập hợp $r_1,r_2,...,\, r_{\phi(m)}$ sắp xếp theo một thứ tự nào đó. Ta có:

$$ar_1ar_2...a\,r_{\phi(m)} \equiv r_1r_2...\,r_{\phi(m)} \ (mod\ m).$$

Như vậy,

$$a^{\phi(m)}\ r_1,r_2,...,r_{\phi(m)} \equiv r_1r_2...\,r_{\phi(m)} \ (mod\ m).$$

Từ đó suy ra định lí.

Định lí Euler có thể dùng để tìm nghịch đảo modulo m. Chẳng hạn nếu a và m là các số nguyên tố cùng nhau, ta có $a.a^{\phi(m)-1} \equiv 1(\text{mod } m)$, tức là $a^{\phi(m)-1}$ chính là nghịch đảo của a modulo m. Từ đó cũng suy ra nghiệm của phương trình đồng dư tuyến tính $ax \equiv b(mod\ m)$, với $(a,m)=1$ là $x \equiv a^{\phi(m)-1} b(mod\ m)$.

Định lí 3.6. *Phi hàm Euler là hàm nhân tính.*

Chứng minh. Giả sử m, n là hai số dương nguyên tố cùng nhau. Ta cần chứng tỏ rằng ϕ(mn)= ϕ(m)ϕ(n). Ta sắp xếp tất cả các số nguyên dương không vượt quá nm thành bảng sau:

1	m+1	2m+1	...	(n-1)m+1
2	m+2	2m+2	...	(n-1)m+2

... ...

r m+r 2m+r ... (n-1)m+r

... ...

m 2m 3m ... mn

Giả sử r là số nguyên không vượt quá m, và $(m,n)=d>1$. Khi đó trong hàng thứ r không có số nào nguyên tố cùng nhau với mn. Vì thế để tính $\phi(mn)$, ta chỉ cần quan tâm các số trong hàng thứ r với $(r,m)=1$. Các số trong hàng này đều nguyên tố cùng nhau với m. Mặt khác dễ thấy rằng các số trong hàng này lập thành một hệ thặng dư đầy đủ modulo n. Do đó có đúng $\phi(n)$ số trong hàng nguyên tố cùng nhau với n, tức là trong hàng có $\phi(n)$ số nguyên tố cùng nhau với mn. Cả thảy có $\phi(n)$ hàng như vậy, định lí được chứng minh.

Nhờ tính chất này ta có ngay công thức Phi-hàm Euler.

Định lí 3.7. *Giả sử $n=p_1^{a1}p_2^{a2}...p_k^{ak}$ là phân tích của n thành thừa số nguyên tố. Khi đó ta có:*

$$\phi(n)=n(1-\frac{1}{p_1})(1-\frac{1}{p_2})...(1-\frac{1}{p_k})$$

Chứng minh. Do Phi-hàm Euler là hàm nhân tính nên ta chỉ cần chứng minh rằng, với mọi số nguyên tố p, $\phi(p^k)=p^k-p^{k-1}$.

Thật vậy, các số nguyên dương không vượt quá p^k và không nguyên tố cùng nhau với p phải có dạng sp với s nguyên dương nào đó. Có đúng p^{k-1} số như vậy. Do đó, số các số không vượt quá p^k và nguyên tố cùng nhau với p^k đúng bằng p^k-p^{k-1}. Tính chất quan trọng sau đây của Phi-hàm thường dược sử dụng về sau.

Định lí 3.8. *Giả sử n là một số nguyên dương. Khi đó*

$$\sum_{d|n}\phi(d)=n$$

trong đó tổng được lấy theo mọi ước của n.

Chứng minh. Ta phân các số nguyên từ 1 đến n thành từng nhóm C_d: $m \in C_d$ khi và chỉ khi $(m,n)=d$, tức là khi và chỉ khi $(m/d, n/d)=1$. Như vậy, số phần tử của C_d đúng bằng số các số nguyên không vượt quá n/d và nguyên tố cùng nhau với n/d, tức là bằng $\phi(n/d)$. Ta có

$$n=\sum_{d|n}\phi(n/d)$$

Khi d chạy qua mọi ước của n thì n/d cũng chạy qua mọi ước của n: định lí được chứng minh.

Nhận xét. Các tính chất của Phi-hàm Euler được sử dụng để tính đồng dư của những luỹ thừa rất lớn. Chẳng hạn, ta cần tính $a^n \bmod k$, trong đó n là một số nguyên lớn. Giả sử ta có

$$k= p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s} .$$

Khi đó $a^{\phi(p_i^{\alpha_i})} \equiv 1 (\mathrm{mod}\ p_i^{\alpha_i})$. Nếu N là bội chung nhỏ nhất của các $\phi(p_i^{\alpha_i})$ thì $a^N \equiv 1 (\mathrm{mod}\ k)$. Do đó, viết $n=Nq+r$ với $r<N$, ta được $a^n \equiv a^r (\mathrm{mod}\ k)$.

Ta xét một ví dụ bằng số. Tính $2^{1000000} \bmod 77$. Ta có: $77=11.7$, $\phi(7)=6$, $\phi(11)=10$. Bội chung nhỏ nhất của 6 và 10 là 30. Ta có $2^{30} \equiv 1 (\mathrm{mod}\ 77)$. Mặt khác, $1000000=30.33333+10$. Vậy

$$2^{1000000} \equiv 2^{10} \equiv 23 (\mathrm{mod}\ 77).$$

§3. Số hoàn hảo và số nguyên tố Mersenne.

Tiết này dành để mô tả một dạng đặc biệt của số nguyên tố, có vai trò quan trọng trong lí thuyết và ứng dụng.

Ta bắt đầu bằng một số hàm số học quan trọng.

Định nghĩa 3.9. Hàm τ(n), số các ước, có giá trị tại n bằng số các ước dương của n; hàm $\sigma(n)$, tổng các ước, có giá trị tại n bằng tổng các ước dương của n. Nói cách khác, ta có:

$$\tau(n)= \sum_{d|n} 1,$$

$$\sigma(n)= \sum_{d|n} d .$$

Ví dụ, nếu p là một số nguyên tố thì $\tau(p)=2$, $\sigma(p)=p+1$.

Định lí 3.10. *$\tau(n)$ và $\sigma(n)$ là các hàm nhân tính.*

Dễ thấy rằng, định lí trên suy ra từ bổ đề sau.

Bổ đề 3.11. *Nếu f là hàm nhân tính, thì $F(n)= \sum_{d|n} f(d)$ cũng là hàm nhân tính.*

Thật vậy, giả sử m, n là các số nguyên dương nguyên tố cùng nhau. Ta có:

$$F(mn)= \sum_{d|mn} f(d) .$$

Vì $(m,n)=1$, mỗi ước d của mn có thể viết duy nhất dưới dạng $d=d_1d_2$ trong đó d_1,d_2 tương ứng là ước của m,n, và d_1,d_2 nguyên tố cùng nhau. Do đó ta có

$$F(mn)= \sum_{d_1|m, d_2|n} f(d_1 d_2)$$

Vì f là hàm nhân tính và $(d_1,d_2)=1$ nên:

$$F(mn)=\sum f(d_1)f(d_2)=\sum_{d_1|m} f(d_1)\sum_{d_2|n} f(d_2)=F(n)F(m)$$

Định lí được chứng minh.

Sử dụng định lí trên, ta có công thức sau đây cho các hàm τ(n) và σ(n).

Định lí 3.12. *Giả sử n có phân tích sau đây ra thừa số nguyên tố* $n=p_1^{a1}p_2^{a2}...p_k^{ak}$. *Khi đó ta có:*

$$\sigma(\mathrm{n})=\prod_{\mathrm{j}=1}^{\mathrm{k}}\frac{\mathrm{p}_{\mathrm{j}}^{\mathrm{aj}+1}-1}{p_j-1}$$

$$\tau(n)=(a_1+1)(a_2+1)...(a_k+1)=\prod_{j=1}^{k}(a_j+1)$$

Chúng tôi dành chứng minh này cho độc giả.

Do các quan niệm thần bí, người cổ Hy Lạp quan tâm đến các số nguyên bằng tổng tất cả các ước dương thực sự của nó. Họ gọi các số đó là các *số hoàn hảo*.

Định nghĩa 3.13. Số nguyên dương *n* được gọi là *số hoàn hảo* nếu $\sigma(n)=2n$.

Ví dụ. Các số 6, 28 là các số hoàn hảo: $\sigma(6)=1+2+3+6=12$, $\sigma(12)=1+2+4+7+14+28=56$
Định lí sau đây được biết từ thời Hy lạp.

Định lí 3.14. *Số nguyên dương chẵn n là số hoàn hảo khi và chỉ khi* $n=2^{m-1}(2^m-1)$, *trong đó m là một số nguyên sao cho* $m\geq 2$ *và* 2^m-1 *là nguyên tố.*

Chứng minh. Trước tiên, giả sử rằng, *m* có dạng như trên. Vì σ là hàm nhân tính, ta có: $\sigma(\mathrm{n})=\sigma(2^{\mathrm{m-1}})\,\sigma(2^{\mathrm{m}}-1)$. Từ công thức của hàm σ và giả thiết 2^m-1 là nguyên tố, dễ thấy rằng $\sigma(2^{\mathrm{m-1}})=2^{\mathrm{m}}-1$, $\sigma(2^{\mathrm{m}}-1)=2^{\mathrm{m}}$, và do đó $\sigma(n)=2n$.

Ngược lại, giả sử *n* là số hoàn hảo chẵn. Viết $n=2^st$, trong đó *s,t* là các số nguyên dương, *t* lẻ, ta được:

$$\sigma(n)=\sigma(2^st)=\sigma(2^s)\,\sigma(t)=(2^{s+1}-1)\,\sigma(t)$$

Vì *n* là số hoàn hảo, $\sigma(n)=2n=2^{s+1}t$.

Như vậy, $2^{s+1}|\sigma(t)$, giả sử $\sigma(t)=2^{s+1}q$. Ta có đẳng thức

$$(2^{s+1}-1)2^{s+1}q=2^{s+1}t,$$

tức là $q|t$ và $q\neq t$. Mặt khác ta có:

$$t+q=(2^{s+1}-1)q+q=2^{s+1}q=\sigma(t)$$

Ta chứng tỏ rằng, $q=1$. Thật vậy, nếu ngược lại, *t* có ít nhất 3 ước khác nhau là *1, t, q,* do đó $\sigma(t)\geq t+q+1$, mâu thuẫn đẳng thức vừa chứng minh. Vậy $\sigma(t)=t+1$, nghĩa là *t* là số nguyên tố. Định lí được chứng minh.

Như vậy để tìm các số hoàn hảo, ta cần tìm các số nguyên tố dạng 2^m-1.

Định nghĩa 3.15. Giả sử m là một số nguyên dương, khi đó $M_m=2^m-1$ được gọi là *số Mersenne thứ m*. Nếu p là số nguyên tố, và M_p cũng nguyên tố, thì M_p được gọi là số nguyên tố Mersenne.

Ví dụ. M_2,M_3,M_5,M_7 là các số nguyên tố Mersenne, trong khi M_{11} là hợp số. Có nhiều định lí khác nhau dùng để xác định số nguyên tố Mersenne. Chẳng hạn nhờ định lí sau đây, ta có thể kiểm tra nhanh chóng dựa vào dạng của các ước số của số nguyên tố Mersenne.

Định lí 3.16. *Nếu p là một số nguyên tố lẻ, thì mọi ước của số nguyên tố Mersenne M_p đều có dạng 2kp+1, trong đó k là số nguyên dương.*

Chứng minh. Giả sử q là một số nguyên tố của M_p. Theo định lí Fermat bé, $q|(2^{q-1}-1)$. Theo hệ quả 1.9, $(2^p-1,2^{q-1}-1)=2^{(p,q-1)}-1$. Ước chung này lớn hơn 1, vì nó là một bội của q. Do đó, $(p,q-1)=p$, vì p là một số nguyên tố. Ta có $q=mp+1$, và vì q lẻ nên $m=2k$, định lí được chứng minh.

Sau đây là vài ví dụ cho thấy ứng dụng của định lí trên.

Ví dụ 1. Để xét xem $M_{13}=2^{13}-1=8191$ có phải là số nguyên tố hay không, ta cần xem các phép chia cho những số nguyên tố không vượt quá $\sqrt{8191}=90,504...$ Mặt khác, theo định lí trên, mọi ước nguyên tố đều phải có dạng $26k+1$. Như vậy chỉ cần thử với hai số 53 và 79: ta thấy M_{13} là số nguyên tố.

Ví dụ 2. Xét $M_{23}=8388607$. Ta cần xét các phép chia của nó cho các số nguyên tố dạng $46k+1$. Số đầu tiên 47 là ước của nó: M_{23} là hợp số.

Có nhiều thuật toán đặc biệt để kiểm tra nguyên tố các số Mersenne. Nhờ đó, người ta phát hiện được những số nguyên tố rất lớn. Mỗi lần có một số nguyên tố Mersenne, ta lại được một số hoàn hảo. Cho đến nay, người ta đã biết được rằng, với $p\leq 132049$, chỉ có 30 số nguyên tố Mersenne, và tính được chúng. Số nguyên tố Mersenne tìm được gần đây nhất là số M_{216091}, gồm 65050 chữ số.

Giả thuyết sau đây vẫn còn chưa được chứng minh.

Giả thuyết 3.17. *Tồn tại vô hạn số nguyên tố Mersenne*.

Người ta đã biết được rằng, trong khoảng từ 1 đến 10^{200} không có số hoàn hảo lẻ. Tuy nhiên câu hỏi sau đây vẫn chưa được trả lời.

Câu hỏi 3.18. *Tồn tại hay không các số hoàn hảo lẻ?*

§4. Căn nguyên thuỷ.

Khi xét các số phức là căn bậc n của đơn vị, ta thường chú ý những số nào không phải là căn của đơn vị với bậc thấp hơn. Những số đó gọi là căn nguyên thuỷ của đơn vị. Đối với các số nguyên, ta cũng có khái niệm hoàn toàn tương tự về "căn" và "căn nguyên thuỷ" của đơn vị.

Định nghĩa 3.19. Giả sử a và m là các số nguyên dương nguyên tố cùng nhau. Khi đó số nguyên nhỏ nhất x thoả mãn đồng dư $a^x \equiv 1 (mod\ m)$ được gọi là *bậc của a modulo m*. Ta viết $x = ord_m a$.

Ta chú ý rằng, số x như vậy tồn tại vì theo định lí Euler, $a^{\phi(m)} \equiv 1 (\mathrm{mod}\ m)$.

Định lí 3.20. *Giả sử a và n là các số nguyên tố cùng nhau, n>0. Khi đó số nguyên x là nghiệm của đồng dư $a^x \equiv 1 (mod\ m)$ khi và chỉ khi x là một bội của bậc của a modulo n.*

Chứng minh. Giả sử x thoả mãn đồng dư trên. Ta viết $x = q\ ord_n a + r$, trong đó $0 \le r < x$. Từ đó ta có $a^r \equiv 1 (mod\ m)$. Vì $ord_n a$ là số dương nhỏ nhất có tính chất đó nên $r = 0$: x là một bội của bậc của a modulo n. Điều ngược lại là rõ ràng.

Hệ quả 3.21. *Nếu a và n là các số nguyên tố cùng nhau, n>0, thì $ord_n a \mid \phi(n)$.*

Hệ quả 3.22. *Nếu a và n là các số nguyên tố cùng nhau, n>0, thì $a^i \equiv a^j (mod\ n)$ khi và chỉ khi $i \equiv j (mod\ n)$.*

Chứng minh các hệ quả trên được dành cho độc giả.

Do hệ quả 3.21, nếu r và n là nguyên tố cùng nhau thì bậc của r không vượt quá $\phi(n)$. Các số có bậc đúng bằng $\phi(n)$ giữ vai trò quan trọng trong nhiều vấn đề khác nhau của số học. Ta có định nghĩa sau.

Định nghĩa 3.23. Nếu r và n là các số nguyên tố cùng nhau, $n>0$, và nếu $ord_n r = \phi(n)$ thì r được gọi là căn nguyên thuỷ modulo n.

Chú ý rằng không phải mọi số đều có căn nguyên thuỷ. Chẳng hạn, xét $n=8$. Các số nhỏ hơn 8 và nguyên tố cùng nhau với 8 là 1, 3, 5, 7, đồng thời ta có $ord_8 1 = 1$, bậc của các số còn lại bằng 2, trong khi $\phi(8) = 4$. Vấn đề những số nguyên nào thì có căn nguyên thuỷ sẽ được xét về sau.

Định lí 3.24. *Nếu r, n nguyên tố cùng nhau, n>0, và nếu r là căn nguyên thuỷ modulo n, thì các số sau đây lập thành hệ thặng dư thu gọn modulo n:*

$$r^1, r^2, \ldots, r^{\phi(n)}.$$

Chứng minh. Vì $(r,n) = 1$, các số trên nguyên tố cùng nhau với n. Ta chỉ cần chứng tỏ rằng, không có hai số nào đồng dư với nhau modulo n. Giả sử $r^i \equiv r^j (mod\ n)$. Theo hệ quả 3.22, $i \equiv j (mod\ \phi(n))$. Từ đó suy ra $i=j$, vì i, j không vượt quá $\phi(n)$. Định lí được chứng minh.

Định lí 3.25. *Nếu $ord_m a = t$ và u là số nguyên dương, thì $ord_m(a^u) = t / (t,u)$.*

Chứng minh. Đặt $v = (t,u)$, $t = t_1 v$, $u = u_1 v$, $s = ord_m(a^u)$. Ta có

$$(a^u)^{t_1} = (a^{u_1 v})^{t/v} = (a^t)^{u_1} \equiv 1 (mod\ m).$$

Do đó, $s \mid t_1$. Mặt khác, $(a^u)^s = a^{us} \equiv 1 (mod\ m)$ nên $t \mid su$. Như vậy, $t_1 v \mid u_1 v s$, do đó, $t_1 \mid u_1 s$. Vì $(u_1, t_1) = 1$, ta có $t_1 \mid s$. Cuối cùng, vì $s \mid t_1$, $t_1 \mid s$ nên $s = t_1 = t/v = t/(t, u)$, chứng minh xong.

Hệ quả 3.26. *Giả sử r là căn nguyên thủy modulo m, trong đó m là số nguyên lớn hơn 1. Khi đó r^u là căn nguyên thủy modulo m nếu và chỉ nếu $(u, \phi(m))=1$.*

Thật vậy, $ord_m r^u = ord_m r/(u, ord_m r) = \phi(m)/(u, \phi(m))$: hệ quả được chứng minh.

Định lí 3.27. *Nếu số nguyên dương m có căn nguyên thuỷ, thì nó có tất cả $\phi(\phi(m))$ căn nguyên thuỷ không đồng dư nhau.*

Thật vậy, nếu *r* là một căn nguyên thuỷ thì $r, r^2, ..., r^{\phi(m)}$ là một hệ đầy đủ các thặng dư thu gọn modulo *m*. Số căn nguyên thuỷ modulo *m* đúng bằng số các số *u* thoả mãn $(u, \phi(m))=1$, và có đúng $\phi(\phi(m))$ số *u* như thế. Định lí được chứng minh.

§5. Sự tồn tại của căn nguyên thuỷ.

Trong tiết này, ta sẽ xác định những số nguyên có căn nguyên thuỷ. Trước tiên ta sẽ chứng minh rằng mọi số nguyên tố đều có căn nguyên thuỷ. Để làm việc đó, ta cần một vài kiến thức về đồng dư đa thức.

Giả sử $f(x)$ là đa thức với hệ số nguyên. Số *c* được gọi là *nghiệm của đa thức f(x) modulo m* nếu $f(c) \equiv 0 (mod\ m)$. Dễ thấy rằng, nếu *c* là một nghiệm thì mọi số đồng dư với *c* modulo *m* cũng là nghiệm.

Đối với số nghiệm của một đa thức modulo một số nguyên, ta cũng có tính chất tương tự như số nghiệm của một đa thức.

Định lí Lagrange. *Giả sử $f(x)=a_n x^n+...+a_1 x+a_0$ là đa thức với hệ số nguyên, n>o, đồng thời $a_n \not\equiv 0 (mod\ p)$. Khi đó f(x) có nhiều nhất n nghiệm modulo p không đồng dư từng cặp.*

Chứng minh. Ta chứng minh bằng qui nạp. Khi $n=1$, định lí là rõ ràng. Giả sử định lí đã chứng minh với đa thức bậc $n-1$ có hệ số của luỹ thừa cao nhất không chia hết cho *p*, và giả sử rằng đa thức $f(x)$ có $n+1$ nghiệm modulo *p* không đồng dư từng cặp $c_0, c_1, ..., c_n$. Ta có $f(x)-f(c_0)=(x-x_0)g(x)$, trong đó $g(x)$ là đa thức bậc $n-1$ với hệ số cao nhất là a_n. Vì với mọi k, $0 \le k \le n$, $c_k - c_0 \not\equiv 0 \pmod p$, trong khi đó $f(c_k)-f(c_0)=(c_k-c_0)g(c_k) \equiv 0 (mod\ p)$, nên c_k là nghiệm của $g(x)$ modulo *p*: trái với giả thiết quy nạp. Định lí được chứng minh.

Định lí 3.28. *Giả sử p là số nguyên tố và d là một ước của p-1. Khi đó đa thức x^d-1 có đúng d nghiệm modulo p không đồng dư từng cặp.*

Chứng minh. Thật vậy, giả sử $p-1=de$. Ta có $x^{p-1}-1=(x^d-1)g(x)$. Theo định lí Fermat bé, $x^{p-1}-1$ có $p-1$ nghiệm modulo *p* không đồng dư từng cặp. Mặt khác, mỗi một nghiệm đó phải là nghiệm của x^d-1 hoặc là của $g(x)$. Theo định lí Lagrange, $g(x)$ có nhiều nhất $p-d-1$ nghiệm không đồng dư từng cặp, vì thế x^d-1 phải có ít nhất $(p-1)-(p-d-1)=d$ nghiệm. Lại theo định lí Lagrange, x^d-1 có không quá *d* nghiệm, vậy nó có đúng *d* nghiệm modulo *p* không đồng dư từng cặp. Định lí dược chứng minh.

Định lí trên đây sẽ được sử dụng trong chương 5 khi xây dựng các trường hữu hạn.

Định lí 3.29. *Giả sử p là số nguyên tố, d là ước dương của p-1. Khi đó, số các số nguyên không đồng dư bậc d modulo p là* $\phi(d)$.

Chứng minh. Giả sử *F(d)* là số các số nguyên dương bậc *d* modulo *p* và bé hơn *p*. Ta cần chứng tỏ rằng $F(d)=\phi(d)$. Vì ϕ(d)=p-1 nên $d|p-1$, từ đó ta có

$$p-1=\sum_{d|p-1}F(d)$$

Mặt khác ta có:

$$p-1=\sum_{d|p-1}\phi(d)$$

theo công thức của Phi-hàm. Như vậy định lí sẽ được chứng minh nếu ta chứng tỏ được rằng $F(d)\le\phi(d)$ nếu $d|p-1$.

Khi $F(d)=0$, điều nói trên là tầm thường. Giả sử $F(d)\neq 0$, tức là tồn tại số nguyên *a* bậc *d* modulo *p*. Khi đó, các số nguyên $a, a^2,...,a^d$ không đồng dư modulo *p*. Rõ ràng rằng, mỗi luỹ thừa của *a* là một nghiệm của $x^d-1\equiv 0(mod\ p)$, mà số nghiệm không đồng dư đúng bằng *d*, nên mỗi nghiệm modulo *p* đồng dư với một trong các luỹ thừa của *a*. Do đó, vì phần tử tuỳ ý bậc *d* là một nghiệm của phương trình $x^d-1\equiv 0(mod\ p)$ nên phải đồng dư với một trong các luỹ thừa của *a*. Mặt khác, theo định lí 3.24, luỹ thừa *k* của *a* có bậc *d* khi và chỉ khi $(k,d)=1$. Có đúng ϕ(d) số *k* như vậy, và do đó suy ra F(d))$\le\phi$(d), định lí được chứng minh.

Hệ quả 3.30. *Mọi số nguyên tố đều có căn nguyên thuỷ.*

Thật vậy, giả sử *p* là số nguyên tố. Khi đó có $\phi(p-1)$ số nguyên bậc *p-1* modulo *p* (Định lí 3.28) không đồng dư từng cặp. Theo định nghĩa, mỗi số đó là một căn nguyên thuỷ: *p* có $\phi(p-1)$ căn nguyên thuỷ.

Phần còn lại của chương được giành để tìm tất cả các số nguyên dương có căn nguyên thuỷ.

Định lí 3.31. *Nếu p là một số nguyên tố lẻ với căn nguyên thuỷ r, thì hoặc r, hoặc r+p là căn nguyên thuỷ modulo* p^2.

Chứng minh. Vì *r* là căn nguyên thuỷ modulo *p* nên ta có

$$ord_p r=\phi(d)=p-1.$$

Giả sử $n=ord_{p^2}\ r$. Ta có $r^n\equiv 1(mod\ p^2)$, và do đó $r^n\equiv 1(mod\ p)$. Như vậy, bậc *p-1* của *r* là một ước của *n*. Mặt khác, *n* là bậc của *r* modulo p^2 nên *n* là ước của $\phi(p^2)=p(p-1)$. Vì $n|p(p-1)$ và $p-1|n$ nên dễ dàng suy ra rằng, hoặc $n=p-1$, hoặc $n=p(p-1)$. Nếu $n=p(p-1)$ thì *r* là căn nguyên thuỷ modulo p^2, vì ord_{p^2} r=ϕ(p^2). Trong trường hợp còn lại, $n=p-1$, ta có $r^{p-1}\equiv 1(\text{mod } p^2)$. Đặt $s=r+p$. Cần phải chứng minh rằng *s* là căn nguyên thuỷ modulo p^2. Vì $s\equiv r(mod\ p)$, *s* cũng là căn nguyên thuỷ

modulo p. Như vậy, theo chứng minh trên ord_{p^2} s hoặc bằng $p-1$, hoặc bằng $p(p-1)$. Ta sẽ chứng tỏ rằng, bậc đó không thể là $p-1$. Ta có

$$s^{p-1}=(r+p)^{p-1}\equiv r^{p-1}+(p-1)pr^{p-2}(mod\ p^2)\equiv 1+(p-1)pr^{p-2}\equiv 1-pr^{p-2}(mod\ p^2)$$

Từ đó ta có thể thấy rằng, $s^{p-1}\not\equiv 1(mod\ p^2)$. Thật vậy, nếu ngược lại thì $pr^{p-2}\equiv 0(\text{mod } p^2)$, nên $r^{p-2}\equiv 0(\text{mod } p)$. Điều này không thể có, vì $p\nmid r$ do r là căn nguyên thuỷ modulo p. Như vậy $ord_{p^2} s=p(p-1)=\phi(p^2)$, tức $s=r+p$ là căn nguyên thuỷ modulo p^2.

Bây giờ ta xét luỹ thừa tuỳ ý của số nguyên tố

Định lý 3.32. *Giả sử p là một số nguyên tố lẻ, khi đó p^k có căn nguyên thuỷ với mọi số nguyên dương k. Hơn nữa, nếu n là căn nguyên thuỷ modulo p^2 thì r là căn nguyên thuỷ modulo p^k với mọi số ngyên dương k.*

Chứng minh. Từ Định lí 3.31, p có căn nguyên thuỷ r sao cho đó cũng là căn nguyên thuỷ modulo p^2, và do đó

$$r^{p-1}\not\equiv 1\ (mod\ p^2).$$

Ta sẽ chứng minh r cũng là căn nguyên thuỷ modulo p^k với mọi số nguyên dương k.

Bằng quy nạp có thể thấy rằng

$$r^{p^{k-1}(p-1)}\not\equiv 1\ (\text{mod } p^k) \qquad (*)$$

với mọi số nguyên dương k. Giả sử

$$n=ord_{p^k} r$$

Ta có $n \mid \varphi(p^k)=p^{k-1}(p-1)$. Mặt khác

$$r^n\not\equiv 1\ (mod\ p^k),$$

và $r^n\not\equiv 1\ (mod\ p)$.

Do đó $p-1=\varphi(p) \mid n$ (Định lí 3.30). Vì $(p-1) \mid n$ và $n \mid p^{k-1}(p-1)$ nên $n=p^t(p-1)$, trong đó t là số nguyên dương $0\le t\le k-1$. Nếu $n=p^t(p-1)$ với $t\le k-2$ thì

$$r^{p^{k-2}(p-1)}=(r^{p^t(p-1)})^{p^{k-2-t}}\equiv 1\ (\text{mod } p^k),$$

mâu thuẫn. Vậy $ord_{p^k} r=p^{k-1}(p-1)=\varphi(p^k)$, r cũng là cũng nguyên thuỷcủa p^k.

Chứng minh (*): k=2: đúng. Giả sử (*) đúng với số nguyên dương $k\ge 2$. Khi đó

$$r^{p^{k-2}(p-1)}\not\equiv 1\ (\text{mod } p^k).$$

Vì $(r,p)=1$, ta thấy $(r,p^{k-1})=1$. Do đó, từ Định lí Euler ta có

$$r^{p^{k-2}(p-1)}\equiv r^{\varphi(p^{k-1})}$$

Vậy tồn tại số nguyên d sao cho

$$r^{p^{k-2}(p-1)} = 1 + dp^{k-1},$$

trong đó $p \nmid d$, vì theo giả thiết $r^{p^{k-2}(p-1)} \not\equiv 1 \pmod{p^k}$.

Ta lấy luỹ thừa bậc p của hai vế phương trình trên và nhận được

$$r^{p^{k-1}(p-1)} = (1+dp^{k-1})^p = 1 + p(dp^{k-1}) + \binom{p}{2} p^2 (dp^{k-1})^2 + \ldots + (dp^{k-1})^p$$
$$\equiv 1 + dp^k \pmod{p^{k+1}}.$$

Vì $p \nmid d$ nên ta có

$$r^{p^{k-1}(p-1)} \not\equiv 1 \pmod{p^{k+1}},$$

chứng minh xong.

Ví dụ: r=3 là căn nguyên thuỷ modulo 7^k với mọi số nguyên dương k.

Định lí 3.33: *Nếu số nguyên dương n không phải là luỹ thừa của một số nguyên tố hoặc hai lần luỹ thừa một số nguyên tố, thì n không có căn nguyên thuỷ.*

Chứng minh. Giả sử n là số nguyên dương với phân tích ra thừa số nguyên tố như sau

$$n = p_1^{t_1} p_2^{t_2} \ldots p_m^{t_m}.$$

Giả sử n có căn nguyên thuỷ r, tức là $(n,r)=1$ và $ord_n r = \varphi(n)$. Vì $(r,n)=1$ nên $(r,p^t)=1$ trong đó p^t là một trong các luỹ thừa nguyên tố có mặt trong phân tích trên. Theo Định lý Euler,

$$r^{\varphi(p^t)} \equiv 1 \pmod{p^t}.$$

Giả sử U là bội chung nhỏ nhất của $\varphi(p_1^{t_1}), \varphi(p_2^{t_2}), \ldots, \varphi(p_m^{t_m})$,

$$U = [\varphi(p_1^{t_1}), \varphi(p_2^{t_2}), \ldots, \varphi(p_m^{t_m})].$$

Vì $\varphi(p_i^{t_i}) \mid U$ nên

$$r^U \equiv 1 \pmod{p_i^{t_i}}$$

Với $I=1, 2, \ldots, m$. Do đó

$$ord_n r = \varphi(n) \le U.$$

Mặt khác,

$$\varphi(n) = \varphi(p_1^{t_1} p_2^{t_2} \ldots p_m^{t_m}) = \varphi(p_1^{t_1}) \varphi(p_2^{t_2}) \ldots \varphi(p_m^{t_m}).$$

Từ đó ta có

$$\varphi(p_1^{t_1}) \varphi(p_2^{t_2}) \ldots \varphi(p_m^{t_m}) \le [\varphi(p_1^{t_1}), \varphi(p_2^{t_2}), \ldots, \varphi(p_m^{t_m})],$$

Tức là $\varphi(p_1^{t_1}), \varphi(p_2^{t_2}),\ldots, \varphi(p_m^{t_m})$ phải nguyên tố cùng nhau từng đôi một. Do $\varphi(p^t)=p^{t-1}(p-1)$ nên $\varphi(p^t)$ chẵn nếu p lẻ, hoặc nếu $p=2$ và $t\geq 2$. Vậy, các số $\varphi(p_1^{t_1}), \varphi(p_2^{t_2}),\ldots, \varphi(p_m^{t_m})$ không nguyên tố cùng nhau từng cặp, trừ trường hợp $m=1$ (và do đó n là luỹ thừa của số nguyên tố), hoặc $m=2$ và $n=2p^t$, trong đó p là số nguyên tố lẻ và t là số nguyên dương.

Định lí 3.34: *Nếu p là số nguyên tố lẻ và t là số nguyên dương, thì $2p^t$ có căn nguyên thuỷ. Cụ thể là, nếu r là căn nguyên thuỷ modulo p^t thì r, (tương ứng, $r+p^t$), là căn nguyên thuỷ modulo $2p^t$ khi r lẻ, (tương ứng, khi r chẵn).*

Chứng minh: Giả sử r là căn nguyên thuỷ modulo p^t, khi đó

$$r^{\varphi(p^t)} \equiv 1 \pmod{p^t},$$

và không có luỹ thừa nào nhỏ hơn $\varphi(p^t)$ thoả mãn đồng dư.

Do $\varphi(2p^t)= \varphi(2)\ \varphi(p^t)= \varphi(p^t)$ nên

$$r^{\varphi(2p^t)} \equiv 1 \pmod{p^t}.$$

Khi r lẻ,

$$r^{\varphi(2p^t)} \equiv 1 \pmod{2}.$$

Từ đó ta có $r^{\varphi(2p^t)} \equiv 1 \pmod{2p^t}$. Vì không có luỹ thừa bé hơn của r thoả mãn đồng dư nên r chính là căn nguyên thuỷ của $2p^t$.

Khi r chẵn, $r+p^t$ lẻ. Do đó,

$$(r+p^t)^{\varphi(2p^t)} \equiv 1 \pmod{2}.$$

Vì $r+p^t \equiv r \pmod{p^t}$ nên

$$(r+p^t)^{\varphi(2p^t)} \equiv 1 \pmod{p^t}\cdot$$

Do đó

$$(r+p^t)^{\varphi(2p^t)} \equiv 1 \pmod{2p^t},$$

và vì không có luỹ thừa bé hơn nào của $(r+p^t)$ thoả mãn đồng dư, ta suy ra $r+p^t$ là căn nguyên thuỷ modulo $2p^t$.

Định lí 3.35: *Nếu a là số nguyên lẻ, $k\geq 3$ là số nguyên thì*

$$a^{\varphi(2^k)/2} = a^{2^{k-2}} \equiv 1 \pmod{2^k}.$$

Chứng minh. Ta chứng minh bằng quy nạp. Giả sử a là số nguyên lẻ, $a=2b+1$.Ta có $a^2=4b(b+1)+1$. Vì b hoặc $b+1$ chẵn nên $8 \mid 4b(b+1)+1$, tức là

$$a^2 \equiv 1 \pmod{8}.$$

Như vậy, định lí đúng khi $k=3$. Giả sử

$$a^{2^{k-2}} \equiv 1 \pmod{2^k}$$

Khi đó tồn tại số nguyên d sao cho

$$a^{2^{k-2}} = 1+d.2^k.$$

Từ đó ta có:

$$a^{2^{k-1}} = 1+d.2^{k+1}+d^2.2^{2k},$$

tức là

$$a^{2^{k-1}} \equiv 1 \pmod{2^{k+1}}.$$

Từ định lí trên ta suy ra rằng, các luỹ thừa 2^k với $k \geq 3$ không có căn nguyên thuỷ. Như vậy, trong các luỹ thừa của 2 chỉ có 2 và 4 là có căn nguyên thuỷ. Kết hợp điều này với các Định lí 3.32, 3.33, 3.34, ta có định lí sau đây

Định lí 3.36: *Số nguyên dương n có căn nguyên thuỷ khi và chỉ khi*

$$n=2, 4, p^t, 2p^t,$$

trong đó p là số nguyên tố lẻ, t là số nguyên dương.

Bài tập và tính toán thực hành chương 3

I. Bài tập

3.1. Hàm Möbius được định nghĩa như sau: $\mu(n)=(-1)^k$, nếu n không chia hết cho số chính phương nào khác 1, và k là số các ước nguyên tố của n; $\mu(1)=1$, $\mu(n)=0$ khi n có ước là số chính phương khác 1.

Chứng minh rằng, với mọi $n>1$, $\sum_{d|n}\mu(d)=0$.

3.2 (Biến đổi ngược Möbius). Cho $f(n)$ là một hàm số học. Đặt

$$F(n)=\sum_{d|n} f(d).$$

Chứng minh rằng:

1) $$f(n)=\sum_{d|n}\mu(d)\,F(n/d).$$

2) Nếu f là hàm nhân tính thì F cũng là hàm nhân tính.

3.3. Dùng biến đổi ngược Möbius và công thức $n=\sum_{d|n}\phi(n/d)$, chứng minh rằng

1) $\phi(p^k)=p^k-p^{k-1}$ với p là số nguyên tố.

2) $\phi(n)$ là hàm nhân tính.

3.4. Cho θ là hàm nhân tính và μ là hàm Möbius. Chứng minh rằng, nếu các ước nguyên tố của n là $p_1,p_2,...p_k$ thì

$$\sum_{d|n}\mu(d)\;\theta(d)=(1-\theta(p_1))(1-\theta(p_2))...(1-\theta(p_k))$$

(nếu n=1, ta xem vế phải là 1)

3.5. Hàm $\sigma_k(n)$ (tổng luỹ thừa bậc k của các ước số của n) được định nghĩa như sau:

$$\sigma_k(n)=\sum_{d|n} d^k .$$

1) Cho công thức tính $\sigma_k(p)$ với p là số nguyên tố.

2) Tính $\sigma_k(p^s)$ khi s là số nguyên dương.

3) Chứng minh rằng $\sigma_k(n)$ là hàm nhân tính.

4) Từ đó cho công thức tính $\sigma_k(n)$ khi $n= p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$.

3.6. Tìm tất cả các số tự nhiên n thoả mãn

$$\sigma(n)+ \phi(n)=2n.$$

3.7. Chứng minh rằng n là một hợp số khi và chỉ khi

$$\sigma(n)>n+\sqrt{(n)}.$$

3.8. Chứng minh rằng nếu hai số nguyên có tích các ước số khác nhau thì hai số nguyên đó khác nhau.

3.9.Tính các đồng dư sau đây bằng nhiều phương pháp khác nhau (chẳng hạn bằng phương pháp bình phương liên tiếp hoặc nhờ nhận xét cuối §2):

1. $3^{1000000} \bmod 165$.
2. $5^{1234567} \bmod 221$.
3. $7^{1000000000} \bmod 541$.

3.10. Chứng minh rằng 91 là số giả nguyên tố cơ sở 3 nhưng không giả nguyên tố Euler cơ sở 3, và không là số giả nguyên tố cơ sở 2.

3.11. Cho $f(n)$ là hàm nhân tính giới nội. Chứng minh rằng tổng

$$\sum f(n)/n^s$$

hội tụ tuyệt đối trong nửa mặt phẳng $Re\ s>1$ (trong đó Re là kí hiệu phần thực của một số), và tổng trong miền hội tụ bằng tích vô hạn hội tụ sau đây

$$\prod_{p\in P}(1+f(p)p^{-s}+\dots+f(p^m)p^{-ms}+\dots),$$

(tích được lấy trên tập hợp tất cả các số nguyên tố).

3.12. Chứng minh rằng, nếu f là hàm nhân tính mạnh giới nội thì

$$\sum_{n=1}^{\infty} f(n)/n^s = \prod_{p\in P}\frac{1}{1-f(p)/p^s}.$$

3.13. Chứng minh đẳng thức sau đối với Zeta-hàm Riemann:

$$\zeta(s)=\sum_{n=1}^{\infty}1/n^s=\prod_{p\in P}\frac{1}{1-p^{-s}}.$$

3.14. Chứng minh rằng nếu $n\neq 2,4,p^{\alpha},2p^{\alpha}$, trong đó p là số nguyên tố lẻ thì

$$a^{\phi(n)/2}\equiv 1(\bmod n).$$

3.15. Chứng minh rằng nếu n chia hết cho 24 thì $\sigma(n)$ cũng chia hết cho 24.

3.17. a) Chứng minh rằng nếu p,q là các số nguyên tố lẻ khác nhau thì $n=pq$ là số giả nguyên tố cơ sở 2 khi và chỉ khi $ord_q 2 | p-1$, $ord_p 2 | q-1$.

b) Trong các số sau đây, số nào là số giả nguyên tố cơ sở 2: 871, 1378, 2047, 2813.

3.18. Chứng minh rằng nếu p,q là các số nguyên tố lẻ khác nhau thì $n=pq$ là số giả nguyên tố cơ sở 2 khi và chỉ khi $M_p M_q=(2^p-1)(2^q-1)$ là số giả nguyên tố cơ sở 2.

3.19. a) Chứng minh rằng nếu đa thức $f(x)$ bậc n, hệ số nguyên, có quá n nghiệm modulo p thì mọi hệ số của $f(x)$ đều chia hết cho p.

b) Cho p là một số nguyên tố. Chứng minh rằng mọi hệ số của đa thức

$$f(x)=(x-1)(x-2)...(x-p+1)-x^{p-1}-x+1$$

chia hết cho p.

c) Dùng câu b) để chứng minh định lí Wilson.

3.20. Tìm tất cả các số tự nhiên n sao cho: $\sigma(n)=12, 18, 24, 48, 52, 84$.

3.21. Chứng minh rằng với mọi $k>1$, phương trình $\tau(n)=k$ có vô số nghiệm.

3.22. Tìm n nhỏ nhất để $\tau(n)=1, 2, 3, 6, 14, 100$.

3.23. Tìm căn nguyên thuỷ modulo:

$$11^2, 17^2, 13^2, 19^2, 3^k, 13^k, 11^k, 17^k.$$

3.24. Chứng minh rằng nếu m có căn nguyên thuỷ thì đồng dư $x^2 \equiv 1 (mod\ m)$ chỉ có nghiệm $x=\pm 1 (mod\ m)$.

3.25. Chứng minh rằng mặc dù không tồn tại căn nguyên thuỷ 2^k, $k \geq 3$, mỗi số nguyên lẻ đồng dư với đúng một số nguyên dạng $(-1)^\alpha 5^\beta$, trong đó $\alpha=0$ hoặc 1, β là số nguyên thoả mãn $0 \leq \beta \leq 2^{k-2}-1$.

3.26. Giả sử n là một số có căn nguyên thuỷ. Chứng minh rằng tích của các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n đồng dư (-1) modulo n (khi n là số nguyên tố, ta có định lí Wilson).

3.27. Tìm tất cả các nghiệm của đồng dư sau:

a) $x^2+x+1 \equiv 0 (mod\ 7)$

b) $x^2+5x+1 \equiv 0 (mod\ 7)$

c) $x^2+3x+1 \equiv 0 (mod\ 7)$.

II. Thực hành tính toán trên máy tính

II. 1. Tính Phi-hàm Euler

Để tính Phi-hàm Euler của một số nguyên dương n ta thực hiện dòng lệnh như sau:

```
[> phi(n);
```

Sau dấu (;) ấn phím "Enter" màn hình sẽ hiện ra kết quả.

Thí dụ: Tính Phi-hàm Euler của 65.

```
[> phi(65);
```

```
                                48
```

II. 2. Thực hành tìm các số khi biết phi-hàm Euler của nó

Để tìm các số khi biết Phi-hàm Euler *k* ta thực hiện dòng lệnh sau:

```
[>invphi(k);
```

Sau dấu (;) ấn phím "Enter" màn hình sẽ hiện ra các số cần tìm.

Thí dụ: Tìm các số khi biết Phi-hàm Euler của nó là 4.

Ta thực hiện như sau:

```
[> invphi(4);
```

```
                                [5, 8, 10, 12]
```

Vậy các số có Phi-hàm Euler bằng 4 là 5, 8, 10, 12.

II. 3. Thực hành kiểm tra số nguyên tố Mersenne

Cho *m* là một số nguyên dương, đặt $M_m:=2^m-1$. Để kiểm tra xem M_m có phải là số nguyên tố Mersenne hay không ta thực hiện dòng lệnh như sau:

```
[> mersenne(m);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình xuất hiện kết quả là một số thì M_m là số nguyên tố Mersenne và M_m chính bằng số đó. Nếu không trên màn hình sẽ xuất hiện chữ "false".

Thí dụ 1: M_7 có phải là số nguyên tố Mersenne hay không?

Ta thực hiện dòng lệnh như sau:

```
[> mersenne(7);
```

```
                                127
```

Vậy M_7=127 và là số nguyên tố Mersenne.

Thí dụ 2: M_{125} có phải là số nguyên tố Mersenne hay không?

```
[> mersenne(125);
```

```
                               false
```

Vậy M_{125} không phải là số nguyên tố Mersenne.

Thí dụ 3: M_{11} có phải là số nguyên tố Mersenne hay không?

```
[> mersenne(11);
```

```
false
```

Vậy M_{11} không phải là số nguyên tố Mersenne.

II. 4. Tính bậc của một số theo một modulo nào đó

Cho *m* là một số nguyên dương, *n* là một số nguyên. Để tính bậc của *n* modulo *m* ta thực hiện dòng lệnh như sau:

```
[>order(n,m);
```

Sau dấu (;) ấn phím "Enter". Nếu *m, n* là các số nguyên tố cùng nhau thì trên màn hình sẽ xuất hiện kết quả chính là bậc của *n* theo modulo *m*. Nếu *m, n* không nguyên tố cùng nhau thì trên màn hình sẽ xuất hiện chữ "FAIL".

Thí dụ 1: Tính bậc của 13 theo modulo 100.

```
[> order(13,100);
```

```
20
```

Vậy $ord_{100}13=20$.

Thí dụ 2: Tính bậc của 5 theo modulo 8

```
[>order(5,8);
```

```
2
```

Vậy $ord_8 5 = 2$.

Thí dụ 3: Tính bậc của 8 theo modulo 12.

```
[> order(8,12);
```

```
FAIL
```

II. 5. Tìm căn nguyên thuỷ

1. Cho *n* là một số nguyên lớn hơn 1. Để tìm căn nguyên thuỷ đầu tiên modulo *n* ta thực hiện dòng lệnh như sau:

```
[> primroot(n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện ra kết quả là một số thì số đó chính là căn nguyên thuỷ đầu tiên modulo *n*. Nếu màn hình hiện ra chữ "FAIL" thì *n* không có căn nguyên thuỷ.

Thí dụ 1: Tìm căn nguyên thuỷ modulo 41.

```
[> primroot(41);
```

```
6
```

Vậy 6 là căn nguyên thuỷ modulo 41.

Thí dụ 2: Tìm căn nguyên thuỷ modulo 15.

```
[> primroot(15);
```

```
                              FAIL
```

Vậy 15 không có căn nguyên thuỷ.

2. Để tìm căn nguyên thuỷ modulo *n* lớn hơn *g* ta thực hiện dòng lệnh sau:

```
[> primroot(g,n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện ra kết quả là một số thì số đó chính là căn nguyên thuỷ lớn hơn *g* đầu tiên modulo *n*. Nếu màn hình hiện ra chữ "FAIL" thì *n* không có căn nguyên thuỷ. Chú ý, nếu *g*=0 thì hai lệnh trên là như nhau.

Thí dụ 1: Tìm căn nguyên thuỷ đầu tiên lớn hơn 7 modulo 41.

```
[> primroot(7,41);
```

```
                               11
```

Vậy 11 là căn nguyên thuỷ lớn hơn 7 đầu tiên modulo 41.

Thí dụ 2: Tìm căn nguyên thuỷ đầu tiên lớn hơn 2 modulo 8.

```
[> primroot(2,8);
```

```
                                FAIL
```

Vậy 8 không có căn nguyên thuỷ lớn hơn 2.

II. 6. Thực hành tính hàm $\tau(n)$

Để tính giá trị của hàm $\tau(n)$ tại *n* ta thực hiện dòng lệnh như sau:

```
[> tau(n);
```

Sau dấu (;) ấn phím "Enter" màn hình sẽ hiện ra kết quả.

Thí dụ 1: Tính $\tau(-9)$.

```
[> tau(-9);
```

```
                               3
```

Thí dụ 2: Tính $\tau(100)$.

```
[> tau(100);
```

```
                               9
```

Vậy số các ước dương của 100 là 9.

II. 7. Thực hành tính hàm $\sigma(n)$

Để tính giá trị của hàm $\sigma(n)$ tại *n* ta thực hiện dòng lệnh như sau:

```
[>sigma(n);
```

Sau dấu (;) ấn phím "Enter" màn hình sẽ hiện ra kết quả.

Thí dụ: Tính $\sigma(9)$.

```
[>sigma(9);
```

13

Vậy tổng các ước dương của 9 là 13.

II. 8. Thực hành tính đồng dư thức, giải phương trình đồng dư

1. Để tính đồng dư của *a* theo modulo *n* ta thực hiện dòng lệnh như sau:

```
[> a mod n;
```

Sau dấu (;) ấn phím "Enter" màn hình sẽ hiện ra kết quả.

Thí dụ: Tính $5^{1234567} \bmod 221$

```
[> 5&^1234567 mod 221;
```

112

2. Để giải phương trình đồng dư ta thực hiện dòng lệnh như sau:

```
[>msolve (các phương trình, modulo);
```

Sau dấu (;) ấn phím "Enter", nếu phương trình đồng dư có nghiệm màn hình sẽ hiện ra kết quả.

Thí dụ: Tìm nghiệm của đồng dư sau:

$$x^2+x+1 \equiv 0 \pmod 7$$

```
[>msolve(x^2+x+1=0,7);
```

x=4, x=2

Vậy nghiệm của phương trình là x=2, x=4(mod 7).

Chương 4.

THẶNG DƯ BÌNH PHƯƠNG.

Giả sử p là một số nguyên tố lẻ, a là số nguyên tố cùng nhau với p. Vấn đề đặt ra là: *khi nào a là số chính phương modulo p?* Vấn đề này không chỉ có giá trị lí thuyết, mà như ta sẽ thấy về sau, có nhiều ứng dụng quan trọng. Để nghiên cứu vấn đề đặt ra, công cụ quan trọng là các kí hiệu Legendre và Jacobi mà ta sẽ xét trong chương này.

§1. Kí hiệu Legendre.

Định nghĩa 4.1. Giả sử m là số nguyên dương. Số a được gọi là một *thặng dư bình phương của m* nếu $(a,m)=1$ và đồng dư $x^2 \equiv a(mod\ m)$ có nghiệm. Nếu ngược lại, ta nói a là *không thặng dư bình phương* của m.

Ta sẽ chứng tỏ rằng, nếu a là một số nguyên tố lẻ, trong số các số 1, 2, ..., p-1 có đúng một nửa là thặng dư bình phương.

Bổ đề 4.1. *Giả sử p là số nguyên tố lẻ, a là số nguyên không chia hết cho p. Khi đó đồng dư sau đây không có nghiệm, hoặc có đúng hai nghiệm không đồng dư modulo p:*

$$x^2 \equiv a(mod\ p).$$

Chứng minh. Giả sử có nghiệm $x=x_0$. Khi đó, dễ chứng minh rằng $x=-x_0$ là một nghiệm không đồng dư với x_0. Ta sẽ chỉ ra rằng, nghiệm tuỳ ý khác $x=x_1$ đồng dư với x_0 hoặc $-x_0$.

Thật vậy, ta có: $x_0^2 \equiv x_1^2 (mod\ p)$, tức là $x_0^2-x_1^2=(x_0+x_1)(x_0-x_1) \equiv 0(mod\ p)$. Do đó, hoặc $p|x_0+x_1$, hoặc $p|x_0-x_1$, điều phải chứng minh.

Định lí 4.3. *Nếu p là một số nguyên tố lẻ, thì trong các số 1, 2, ..., p-1 có đúng (p-1)/2 thặng dư bình phương.*

Chứng minh. Để tìm tất cả các thặng dư modulo p trong các số 1,2,...,p-1, trước tiên ta bình phương các số đó và xét các thặng dư dương bé nhất modulo p của các kết quả nhận được. Các thặng dư dương bé nhất này là tất cả các thặng dư bình phương trong các số từ 1 đến p-1. Giả sử a là một thặng dư như vậy. Vì phương trình đồng dư $x^2 \equiv a(mod\ p)$ có đúng hai nghiệm, nên trong số $(p-1)$ bình phương đang xét, phải có hai bình phương thặng dư a: Số thặng dư bình phương đúng bằng (p-1)/2.

Để xét các thặng dư bình phương, người ta thường dùng các kí hiệu quan trọng mà ta sẽ nghiên cứu trong chương này.

Định nghĩa 4.4. Giả sử p là một số nguyên tố lẻ và a là một số nguyên không chia hết cho p. *Kí hiệu Legendre* $\left[\frac{a}{p}\right]$ được định nghĩa như sau:

$$\left[\frac{a}{p}\right] = \begin{cases} 1, \text{ nếu } a \text{ là thặng dư bình phương của } p \\ -1, \text{ nếu ngược lại.} \end{cases}$$

Ví dụ. Dễ tính được:

$$\left[\frac{1}{11}\right] = \left[\frac{3}{11}\right] = \left[\frac{4}{11}\right] = \left[\frac{5}{11}\right] = \left[\frac{9}{11}\right] = 1.$$
$$\left[\frac{2}{11}\right] = \left[\frac{6}{11}\right] = \left[\frac{7}{11}\right] = \left[\frac{8}{11}\right] = \left[\frac{10}{11}\right] = -1.$$

Tiêu chẩn sau đây thường được dùng để chứng minh các tính chất của kí hiệu Legendre.

Định lí (Tiêu chuẩn Euler). *Giả sử p là số nguyên tố lẻ, và a là số nguyên dương không chia hết cho p. Khi đó:*

$$\left[\frac{a}{p}\right] \equiv a^{(p-1)/2} (mod\ p).$$

Chứng minh. Trước tiên, giả sử rằng $\left[\frac{a}{p}\right] = 1$. Khi đó, đồng dư $x^2 \equiv a (mod\ p)$ có nghiệm $x = x_0$. Theo định lí Fermat bé, ta có:

$$a^{(p-1)/2} = (x_0^2)^{(p-1)/2} = x_0^{p-1} \equiv 1 (mod\ p)$$

Chỉ còn phải xét trường hợp $\left[\frac{a}{p}\right] = -1$. Khi đó , đồng dư $x^2 \equiv a (mod\ p)$ vô nghiệm. Với mỗi i sao cho $1 \le i \le p-1$, tồn tại duy nhất j $(1 \le j \le p-1)$ để $ij \equiv a (mod\ p)$. Rõ ràng $i \ne j$, nên ta có thể nhóm các số $1, ..., p-1$ thành $(p-1)/2$ cặp với tích từng cặp đồng dư a modulo p. Nhân các cặp này với nhau ta được:

$$(p-1)! \equiv a^{(p-1)/2} (mod\ p).$$

Từ định lí Wilson ta có:

$$-1 \equiv a^{(p-1)/2} (mod\ p).$$

Định lí được chứng minh.

Những tính chất sau đây cho phép tính được dễ dàng kí hiệu Legendre.

Định lí 4.5. *Giả sử p là một số nguyên tố lẻ, a và b là các số nguyên không chia hết cho p. Khi đó:*

(i) Nếu $a \equiv b(\text{mod } p)$ *thì* $\left[\frac{a}{p}\right] = \left[\frac{b}{p}\right]$.

(ii) $\left[\frac{a}{p}\right]\left[\frac{b}{p}\right] = \left[\frac{ab}{p}\right]$.

(iii) $\left[\frac{a^2}{p}\right] = 1$.

Chứng minh. (i). Nếu $a \equiv b(\text{mod } p)$ thì $x^2 \equiv a(\text{mod } p)$ có nghiệm nếu và chỉ nếu $x^2 \equiv b(\text{mod } p)$ có nghiệm. Do đó $\left[\frac{a}{p}\right] = \left[\frac{b}{p}\right]$.

(ii). Bởi tiêu chuẩn Euler ta có:

$$\left[\frac{a}{p}\right] \equiv a^{(p-1)/2}(\text{mod } p), \left[\frac{b}{p}\right] \equiv b^{(p-1)/2}(\text{mod } p).$$

$$\left[\frac{ab}{p}\right] \equiv (ab)^{(p-1)/2}(\text{mod } p).$$

Như vậy,

$$\left[\frac{a}{p}\right]\left[\frac{b}{p}\right] \equiv a^{(p-1)/2}b^{(p-1)/2} = (ab)^{(p-1)/2} \equiv \left[\frac{ab}{p}\right] (\text{mod } p).$$

Vì giá trị của kí hiệu Legendre chỉ có thể là ± 1 nên ta có đẳng thức cần chứng minh.

(iii) Vì $\left[\frac{a}{p}\right] = \pm 1$ nên từ phần trên ta có

$$\left[\frac{a^2}{p}\right] = \left[\frac{a}{p}\right]\left[\frac{a}{p}\right] = 1.$$

Định lí trên cho thấy rằng tích của hai *thặng dư bình phương* hoặc hai *không thặng dư bình phương* là một *thặng dư bình phương*, tích của một *thặng dư bình phương* và một *không thặng dư bình phương* là một *không thặng dư bình phương*.

Tiêu chuẩn Euler cho biết khi nào thì các số nguyên lẻ nhận -1 là thặng dư bình phương.

Định lí 4.6. *Nếu p là số nguyên tố lẻ thì*

$$\left[\frac{-1}{p}\right] = \begin{cases} 1, & khi\ \ p \equiv 1 (\mathrm{mod}\, 4) \\ -1, & khi\ \ p \equiv -1 (\mathrm{mod}\, 4) \end{cases}$$

Chứng minh. Theo tiêu chuẩn Euler ta có:

$$\left[\frac{-1}{p}\right] \equiv (-1)^{(p-1)/2} (\mathrm{mod}\ p).$$

Nếu $p \equiv 1 (mod\ 4)$ thì $p=4k+1$ với k nguyên nào đó. Như vậy,

$$(-1)^{(p-1)/2}=(-1)^{2k+1}=-1,$$

tức là $\left[\frac{-1}{p}\right]=-1$.

Định lí 4.7. (Bổ đề Gauss). *Giả sử p là số nguyên tố lẻ và (a,p)=1. Nếu s là số các thặng dư dương bé nhất của các số nguyên a,2a,...((p-1)/2)a lớn hơn p/2, thì* $\left[\frac{a}{p}\right]=(-1)^s$.

Chứng minh. Trong số các thặng dư dương bé nhất của các số nguyên *a,2a,..., ((p-1)/2)a,* giả sử $u_1,u_2,...,u_s$ là các thặng dư lớn hơn p/2, và $v_1,v_2,...,v_t$ là các thặng dư nhỏ hơn *p/2*. Vì $(ja,p)=1$ với mọi $j,\ 1 \le j \le (p-2)/2$, nên tất cả các thặng dư dương bé nhất nói trên đều nằm trong tập hợp *1,...,p-1*.

Ta sẽ chứng tỏ rằng, $p-u_1,...,\ p-u_s,\ v_1,...,v_t$ chính là tập hợp các số *1,...(p-1)/2,* xếp theo thứ tự nào đó. Có cả thảy *(p-1)/2* số không vượt quá (p-1)/2, nên chỉ còn phải chứng minh rằng không có hai số nào đồng dư với nhau.

Rõ ràng không có hai số u_i nào, cũng như không có hai số v_j nào đồng dư với nhau modulo *p*. Thật vậy, nếu ngược lại, ta sẽ có đồng dư $ma \equiv na (mod\ p)$ với *m, n* dương nào đó không vượt quá *(p-1)/2*. Vì *(a,p)=1* nên từ đó suy ra $m \equiv n (mod\ p)$: Mâu thuẫn.

Tương tự như trên, có thể thấy rằng không có $p-u_i$ nào đó đồng dư với v_j.

Vậy ta có:

$$(p-u_1)...(p-u_s)v_1...v_t \equiv \left[\frac{p-1}{2}\right]!(mod\ p).$$

Từ đó suy ra

$$(-1)^s u_1...u_s v_1...v_t \equiv \left[\frac{p-1}{2}\right]!(mod\ p).$$

Mặt khác, vì $u_1,...u_s,v_1,...v_t$ là các thặng dư dương bé nhất của $a,2a,...,((p-1)/2)a$ nên

$$u_1...u_s v_1...v_t \equiv a^{(p-1)/2}\left[\frac{p-1}{2}\right]!(mod\ p).$$

Như vậy ta có:

$$(-1)^s\ a^{(p-1)/2}\left[\frac{p-1}{2}\right]! \equiv \left[\frac{p-1}{2}\right]!(mod\ p).$$

Vì $(p,((p-1)/2)!)=1$ nên suy ra:

$$(-1)^s a^{(p-1)/2} \equiv 1(mod\ p),$$

tức là:

$$a^{(p-1)/2} \equiv (-1)^s(mod\ p)$$

Định lí suy ra từ tiêu chuẩn Euler.

Định lí 4.8. *Nếu p là một số nguyên tố lẻ thì*

$$\left[\frac{2}{p}\right]=(-1)^{(p^2-1)/8}$$

Như vậy, 2 là thặng dư bình phương của mọi số nguyên tố dạng $p \equiv \pm 1(mod\ 8)$*và là không thặng dư bình phương của mọi số nguyên tố dạng* $p \equiv \pm 3(mod\ 8)$.

Chứng minh. Áp dụng tiêu chuẩn Gauss, ta cần tính số thặng dư dương bé nhất lớn hơn $p/2$ của dãy số

$$1.2,2.2,...,((p-1)/2).2$$

Vì các số đều nhỏ hơn p nên các thặng dư dương bé nhất của mỗi số trùng với chính nó. Như vậy, ta chỉ cần tính số các số của dãy lớn hơn $p/2$. Số các số đó là $s=(p-1)/2-[p/4]$ (trong đó [ ] chỉ phần nguyên). Như vậy ta có:

$$\left[\frac{2}{p}\right]=(-1)^{(p-1)/2-[p/4]}.$$

Dễ kiểm tra đồng dư thức sau đây bằng cách phân ra các trường hợp $p \equiv 1,3,5,7(mod\ 8)$:

$$(p-1)/2-[p/4] \equiv (p^2-1)/8(mod\ 2)$$

Từ đó ta có:

$$\left[\frac{2}{p}\right] \equiv (-1)^{(p^2-1)/8}(mod\ 2).$$

Tính toán trực tiếp cho đẳng thức cần chứng minh.

§2. Luật thuận nghịch bình phương.

Định lí sau đây cho ta mối liên hệ giữa các kí hiệu Legendre $\left[\frac{p}{q}\right]$ và $\left[\frac{q}{p}\right]$. Định lí này thường được sử dụng khi tính toán với các kí hiệu Legendre.

Định lí 4.9. (Luật thuận nghịch bình phương). *Giả sử p và q là các số nguyên tố lẻ, khi đó ta có:*

$$\left[\frac{p}{q}\right]\left[\frac{q}{p}\right]=(-1)^{((p-1)/2).((q-1)/2)}.$$

Trước hết ta chứng minh bổ đề sau.

Bổ đề 4.10. *Giả sử p là một số nguyên tố lẻ, a là một số lẻ không chia hết cho p. Khi đó*

$$\left[\frac{a}{p}\right]=(-1)^{T(a,p)},$$

trong đó

$$T(a,p)=\sum_{j=1}^{(p-1)/2}[ja/p].$$

Chứng minh. Xét các thặng dư dương bé nhất của các số nguyên $a,2a,...,((p-1)/2)a$. Giả sử $u_1,...u_s,v_1,...v_t$ tương ứng là các thặng dư lớn hơn và bé hơn $p/2$. Ta có:

$$ja=p[ja/p]+\text{phần dư}$$

trong đó phần dư là một trong các số u_i hoặc v_j. Cộng từng vế $(p-1)/2$ phương trình, ta được:

$$\sum_{j=1}^{(p-1)/2}ja=\sum_{j=1}^{(p-1)/2}p[ja/p]+\sum_{j=1}^{s}u_j+\sum_{j=1}^{t}v_j$$

Như đã chứng tỏ trong chứng minh bổ đề Gauss, các số nguyên $p-u_1,..., p-u_s, v_1,...,v_t$ chính là tập hợp các số $1,...(p-1)/2$, xếp theo thứ tự nào đó.Vậy ta có:

$$\sum_{j=1}^{(p-1)/2}j=\sum_{j=1}^{s}(p-u_j)+\sum_{j=1}^{t}v_j=ps-\sum_{j=1}^{s}u_j+\sum_{j=1}^{t}v_j$$

Từ đó suy ra

$$\sum_{j=1}^{(p-1)/2}ja-\sum_{j=1}^{(p-1)/2}j=\sum_{j=1}^{(p-1)/2}p[ja/p]-ps+2\sum_{j=1}^{s}u_j$$

Từ công thức của $T(a,p)$, ta nhận được:

$$(a-1)\sum_{j=1}^{(p-1)/2} j = pT(a,p) - ps + 2\sum_{j=1}^{s} u_j \ .$$

Vì a, p lẻ nên

$$T(a,p) \equiv s (mod\ 2)$$

Bổ đề được chứng minh bằng cách áp dụng bổ đề Gauss.

Bây giờ ta chứng minh Luật thuận nghịch bình phương.

Xét các cặp số nguyên (x,y) với $1 \le x \le (p-1)/2$ và $1 \le y \le (q-1)/2$. Có tất cả $((p-1)/2)((q-1)/2)$ cặp như vậy. Ta sẽ chia các cặp đó thành hai nhóm tuỳ thuộc độ lớn của qx và py.

Trước tiên, dễ thấy rằng $qx \ne py$ đối với mọi cặp.

Để đánh số các cặp số nguyên (x,y) với $1 \le x \le (p-1)/2$, $1 \le y \le (q-1)/2$ và $qx > py$, ta chú ý rằng chúng chính là các cặp với $1 \le x \le (p-1)/2$, $1 \le y \le qx/p$. Với mỗi giá trị cố định của x, $1 \le x \le (p-1)/2$, tồn tại $[qx/p]$ số nguyên thoả mãn $1 \le y \le qx/p$. Như vậy số các cặp thoả mãn tính chất đang xét là $\sum_{j=1}^{(p-1)/2} [qj/p]$.

Tiếp theo, ta xét các cặp thoả mãn $1 \le x \le (p-1)/2$, $1 \le y \le (q-1)/2$ và $qx < py$. Lý luận tương tự như trên cho thấy, số các cặp là $\sum_{j=1}^{(q-1)/2} [pj/q]$.

Vì có tất cả là $((p-1)/2)((q-1)/2)$ cặp, ta nhận được đẳng thức sau

$$\sum_{j=1}^{(p-1)/2} [qj/p] + \sum_{j=1}^{(q-1)/2} [pj/q] = ((p-1)/2)((q-1)/2).$$

Từ định nghĩa của hàm T, ta có:

$$(-1)^{T(p,q)+T(q,p)} = (-1)^{((p-1)/2)((q-1)/2)}$$

Định lý được suy ra từ bổ đề 4.10

Nhận xét. Định lý trên đây (Luật thuận nghịch bình phương) thường được dùng để tính ký hiệu Legendre. Chẳng hạn, từ định lí có thể suy ra rằng, $\left[\frac{p}{q}\right]\left[\frac{q}{p}\right] = -1$ nếu $p \equiv q \equiv 3 (mod\ 4)$, và bằng 1 trong các trường hợp còn lại, tức là $\left[\frac{p}{q}\right] = \left[\frac{-q}{p}\right]$ nếu $p \equiv q \equiv 3 (mod\ 4)$, và $\left[\frac{p}{q}\right] = \left[\frac{q}{p}\right]$ trong các trường hợp có ít nhất một trong hai số p hoặc q đồng dư với 1 modulo 4.

Ta xét một ví dụ bằng số: tính $\left[\begin{matrix}713\\1009\end{matrix}\right]$.

$$\left[\begin{matrix}713\\1009\end{matrix}\right]=\left[\begin{matrix}23.31\\1009\end{matrix}\right]=\left[\begin{matrix}23\\1009\end{matrix}\right]\left[\begin{matrix}31\\1009\end{matrix}\right]$$

Vì $1009 \equiv 1 (mod\ 4)$ nên ta có:

$$\left[\begin{matrix}23\\1009\end{matrix}\right]=\left[\begin{matrix}1009\\23\end{matrix}\right],\left[\begin{matrix}31\\1009\end{matrix}\right]\left[\begin{matrix}1009\\31\end{matrix}\right]$$

Mặt khác,

$$\left[\begin{matrix}1009\\23\end{matrix}\right]=\left[\begin{matrix}20\\23\end{matrix}\right]=\left[\begin{matrix}2^2 5\\23\end{matrix}\right]=\left[\begin{matrix}2^2\\23\end{matrix}\right]\left[\begin{matrix}5\\23\end{matrix}\right]=\left[\begin{matrix}5\\23\end{matrix}\right]=\left[\begin{matrix}23\\5\end{matrix}\right]=\left[\begin{matrix}3\\5\end{matrix}\right]=\left[\begin{matrix}5\\3\end{matrix}\right]=\left[\begin{matrix}2\\3\end{matrix}\right]=-1$$

$$\left[\begin{matrix}1009\\31\end{matrix}\right]=\left[\begin{matrix}17\\31\end{matrix}\right]=\left[\begin{matrix}31\\17\end{matrix}\right]=\left[\begin{matrix}14\\17\end{matrix}\right]=\left[\begin{matrix}2\\17\end{matrix}\right]\left[\begin{matrix}7\\17\end{matrix}\right]=\left[\begin{matrix}7\\17\end{matrix}\right]=\left[\begin{matrix}17\\7\end{matrix}\right]=\left[\begin{matrix}3\\7\end{matrix}\right]=-\left[\begin{matrix}7\\3\end{matrix}\right]=-\left[\begin{matrix}4\\3\end{matrix}\right]$$

$$=-\left[\begin{matrix}2^2\\3\end{matrix}\right]=-1$$

Vậy, $\left[\begin{matrix}713\\1009\end{matrix}\right]=1$.

Luật thuận nghịch bình phương còn được dùng trong kiểm tra nguyên tố. Ta có định lí sau.

Định lí 4.11. (Kiểm tra Pepin). *Số Fermat F_m là số nguyên tố khi và chỉ khi*

$$3^{(F_m-1)/2} \equiv -1 (mod\ F_m)$$

Chứng minh. Ta nhắc lại định nghĩa số Fermat: $F_m=2^{2^m}+1$.

Giả sử đồng dư phát biểu trong định lí được thoả mãn. Khi đó ta có

$$3^{F_m-1} \equiv 1 \pmod{F_m}$$

Như vậy, nếu F_m có ước nguyên tố p thì

$$3^{F_m-1} \equiv 1 \pmod{p}$$

Do đó, $ord_p 3$ phải là một ước của F_m-1, tức phải là một luỹ thừa của 2. Từ giả thiết suy ra $ord_p 3 \mid (F_m-1)/2=2^{2^m-1}$. Vậy ta có: $ord_p 3=F_m-1$. Từ đó suy ra $F_m-1 \le p-1$, nhưng vì p là ước của F_m, nên có nghĩa là $F_m=p$: F_m là số nguyên tố.

Ngược lại, giả sử F_m nguyên tố. Theo luật thuận nghịch bình phương, ta có:

$$\left[\frac{3}{F_m}\right] = \left[\frac{F_m}{3}\right] = \left[\frac{2}{3}\right] = -1$$

Mặt khác, theo tiêu chuẩn Euler ta có:

$$\left[\frac{3}{F_m}\right] \equiv 3^{(F_m-1)/2} \pmod{F_m}$$

Định lí đã được chứng minh.

Nhận xét. Dùng tiêu chuẩn Pepin, dễ kiểm tra được rằng F_1,F_2,F_3,F_4 là các số nguyên tố, F_5 là hợp số.

§3. Kí hiệu Jacobi.

Kí hiệu Jacobi là một mở rộng của kí hiệu Legendre, và được sử dụng để tính kí hiệu Legendre, cũng như trong nhiều vấn đề nghiên cứu các số giả nguyên tố.

Định nghĩa 4.12. Giả sử n là số nguyên dương lẻ, a nguyên tố cùng nhau với n. Nếu n có phân tích ra thừa số nguyên tố là $p_1^{t_1} p_2^{t_2} ... p_m^{t_m}$, ta định nghĩa *kí hiệu Jacobi* như sau:

$$\left[\frac{a}{n}\right] = \left[\frac{a}{p_1}\right]^{t_1}\left[\frac{a}{p_2}\right]^{t_2}\left[\frac{a}{p_m}\right]^{t_m},$$

trong đó ở vế phải là các kí hiệu Legendre.

Như vậy, trong trường hợp n là số nguyên tố thì kí hiệu Jacobi trùng với kí hiệu Legendre. Tuy nhiên cần chú ý rằng, khác với kí hiệu Legendre, khi n là hợp số, kí hiệu Jacobi không cho ta biết phương trình đồng dư $x^2 \equiv a \pmod p$ có nghiệm hay không. Mặc dầu vậy, kí hiệu Jacobi có nhiều tính chất tương tự với kí hiệu Legendre.

Định lí 4.13. *Giả sử n là số nguyên dương lẻ, a và b là các số nguyên tố cùng nhau với n. Khi đó:*

(i) Nếu $a \equiv b \pmod n$ *thì* $\left[\frac{a}{n}\right] = \left[\frac{b}{n}\right]$.

(ii) $\left[\frac{ab}{n}\right] = \left[\frac{a}{n}\right]\left[\frac{b}{n}\right]$

(iii) $\left[\frac{-1}{n}\right] = (-1)^{(n-1)/2}$

$(iv)\left[\frac{2}{n}\right]=(-1)^{(n^2-1)/8}$

Chứng minh. Hai đẳng thức đầu tiên dễ suy ra từ định nghĩa kí hiệu Jacobi và tính chất của kí hiệu Legendre.

Để chứng minh tính chất thứ 3, ta nhận xét rằng, do (p_i-1) chẵn nên

$$(1+(p_i-1))^{t1} \equiv 1+t_i(p_i-1)(mod\ 4),$$

$$(1+t_i(p_i-1))(1+t_j(p_j-1)) \equiv 1+t_i(p_i-1)+ t_j(p_j-1)(mod\ 4).$$

Từ đó suy ra:

$$n \equiv 1+t_1(p_1-1)+t_2(p_2-1)+...+ t_m(p_m-1)(mod\ 4),$$

tức là,

$$(n-1)/2 \equiv t_1(p_1-1)/2+t_2(p_2-1)/2+...+ t_m(p_m-1)/2(mod\ 2)$$

Hệ thức này cùng với định nghĩa cho ta đẳng thức (iii).

Chứng minh (iv). Ta có:

$$\left[\frac{2}{n}\right]=\left[\frac{2}{p_1}\right]^{t_1}\left[\frac{2}{p_2}\right]^{t_2}\cdots\left[\frac{2}{p_m}\right]^{t_m}=(-1)^{t_1(p_1^2-1)/8+t_2(p_2^2-1)/8+...+t_m(p_m^2-1)/8}$$

Lập luận tương tự như trong chứng minh phần trên,ta có:

$$n^2 \equiv 1+t_1(p_1^2-1)+t_2(p_2^2-1)+...+ t_m(p_m^2-1)(mod\ 64),$$

và khi đó (iv) suy ra từ định nghĩa.

Định lí 4.14. (Luật thuận nghịch bình phương đối với kí hiệu Jacobi). *Giả sử m, n là các số nguyên dương lẻ, nguyên tố cùng nhau. Khi đó:*

$$\left[\frac{n}{m}\right]\left[\frac{m}{n}\right]=(-1)^{\frac{m-1}{2}\frac{n-1}{2}}.$$

Chứng minh. Giả sử *m, n* có phân tích ra thừa số nguyên tố dạng: $m=p_1^{a_1}p_2^{a_2}...p_s^{a_s}$, $n=q_1^{b_1}q_2^{b_2}...q_r^{b_r}$. Dùng định nghĩa và luật thuận nghịch bình phương của kí hiệu Legendre, ta được:

$$\left[\frac{n}{m}\right]\left[\frac{m}{n}\right]=\prod_{i=1}^{r}\prod_{j=1}^{s}(-1)^{a_j\frac{p_j-1}{2}b_i\frac{q_i-1}{2}}=(-1)^{\sum_{i=1}^{r}\sum_{j=1}^{s}a_j\frac{p_j-1}{2}b_i\frac{q_i-1}{2}}.$$

Như trong chứng minh định lí 4.13, (iii), ta có:

$$\sum_{j=1}^{s} a_j \frac{p_j - 1}{2} \equiv \frac{m-1}{2} (\mathrm{mod}\, 2),$$

$$\sum_{i=1}^{r} b_i \frac{q_i - 1}{2} \equiv \frac{n-1}{2} (\mathrm{mod}\, 2).$$

Từ đó suy ra định lí.

§4. Thuật toán tính kí hiệu Jacobi.

Giả sử *a, b* là hai số nguyên dương nguyên tố cùng nhau, *a>b*. Đặt $R_1=a$, $R_2=b$. Dùng thuật chia Eulid và tách luỹ thừa cao nhất của 2 trong phần dư, ta được:

$$R_0 = R_1 q_1 + 2^{s_1} R_2$$

$$R_1 = R_2 q_2 + 2^{s_2} R_3$$

.....

$$R_{n-3} = R_{n-2} q_{n-2} + 2^{s_{n-2}} R_{n-1}$$

$$R_{n-2} = R_{n-1} q_{n-1} + 2^{s_{n-1}} .1$$

trong đó s_j là các số nguyên không âm, R_j là số nguyên lẻ bé hơn R_{j-1}.

Ta chú ý rằng, số các phép chia đòi hỏi trong thuật toán trên là không vượt quá số phép chia cần thiết khi dùng thuật toán Euclid để tìm ước chung lớn nhất của hai số *a* và *b*.

Đặt:

$$R(a,b) = s_1 \frac{R_1^{\,2} - 1}{8} + s_2 \frac{R_2^{\,2} - 1}{8} + \ldots + s_{n-1} \frac{R_{n-1}^{\,2} - 1}{8} + \frac{R_1 - 1}{2} . \frac{R_2 - 1}{2} + \ldots + \frac{R_{n-2} - 1}{2} . \frac{R_{n-1} - 1}{2}.$$

Ta có định lí sau.

Định lí 4.15. *Giả sử a,b là các số nguyên dương và a>b. Khi đó ta có:*

$$\left[\frac{a}{b}\right] = (-1)^{R(a,b)}$$

Chứng minh. Theo các phần *(i), (ii)*, và *(iv)* của định lí 4.13 ta có:

$$\left[\frac{a}{b}\right] = \left[\frac{R_0}{R_1}\right] = \left[\frac{2^{s_1} R_2}{R_1}\right] = \left[\frac{2}{R_1}\right]^{s_1} \left[\frac{R_2}{R_1}\right] = (-1)^{s_1 \frac{R_1^{\,2}-1}{8}} \left[\frac{R_2}{R_1}\right].$$

Dùng luật thuận nghịch bình phương của kí hiệu Jacobita được:

$$\begin{bmatrix} R_2 \\ R_1 \end{bmatrix} = (-1)^{\frac{R_1-1}{2}\frac{R_2-1}{2}} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}.$$

Như vậy,

$$\begin{bmatrix} a \\ b \end{bmatrix} = (-1)^{\frac{R_1-1}{2}\frac{R_2-1}{2}+s_1\frac{R_1^2-1}{8}} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

Tiếp tục quá trình đó, ta đi đến công thức cần chứng minh.

Hệ quả 4.16. *Giả sử a và b là các số nguyên dương nguyên tố cùng nhau, a>b. Khi đó, kí hiệu Jacobi* $\begin{bmatrix} a \\ b \end{bmatrix}$ *có thể tính được với* $O((\log_2 b)^3)$ *phép tính bit.*

Chứng minh. Như ta đã nhận xét, số các phép chia trong thuật toán xác định *R(a,b)* không vượt quá số phép chia trong thuật toán Euclid để tính ước chung lớn nhất của *a* và *b*. Theo định lí Lamé, cần có $O(\log_2 b)$ phép chia. Mỗi phép chia cần không quá $O((\log_2 b)^2)$ phép tính bit. Sau mỗi phép chia, cặp số R_j, s_j tìm được bởi $O(\log_2 b)$ phép tính bit (chỉ cần là các phép dịch chuyển). Như vậy, khi biết *a, b,* chỉ cần $O((\log_2 b)^3)$ phép tính bit để xác định các số R_j, s_j. Để nâng (-1) lên luỹ thừa *R(a,b)* như trong định lí, ta chỉ cần sử dụng 3 chữ số nhị phân cuối cùng của R_j và chữ số nhị phân cuối cùng của s_j, vì giá trị luỹ thừa của (-1) chỉ phụ thuộc vào tính chẵn lẻ của số mũ. Như vậy, khi đã có R_j, s_j, ta chỉ cần $O(\log_2 b)$ để xác định $\begin{bmatrix} a \\ b \end{bmatrix}$. Hệ quả được chứng minh.

Ta có thuật toán sau đây để tính kí hiệu Jacobi dựa vào các định lí vừa chứng minh.

Thuật toán tính kí hiệu Jacobi $\begin{bmatrix} a \\ b \end{bmatrix}$ (và do đó, tính kí hiệu Legendre khi *b* là số nguyên tố).

J1. (Kiểm tra b≠0). Nếu b=0, in ra 0 nếu |a|≠1, in ra 1 nếu |a|=1 và kết thúc thuật toán.

J2. (Tách các luỹ thừa của 2 khỏi *b*). Nếu *a* và *b* đều chẵn, in ra 0 và kết thúc thuật toán. Ngược lại, đặt v←0, và khi *b* chẵn, đặt v←v+1, b←b/2. Sau đó, nếu *v* chẵn, đặt k←1, ngược lại, đặt k←$(-1)^{(a^2-1)/8}$. Cuối cùng, nếu b<0, đặt b←-b, và nếu hơn nữa, a<0, đặt k←-k.

J3. (Kết thúc?). (ở bước này, ta có *b* lẻ và b>0). Nếu a=0, in ra 0 nếu b>1, in ra *k* nếu b=1 và kết thuật toán. Ngược lại, đặt v←0 và nếu *a* chẵn, đặt v←v+1, a←a/2. Nếu *v* lẻ, đặt k← $(-1)^{(b^2-1)/8}k$.

J4. (Sử dụng luật thuận nghịch). Đặt k←$(-1)^{(a-1)(b-1)/4}$k.

Nhận xét. Ở đây, ta cần lưu ý một điều. Mặc dù trong thuật toán có xuất hiện các phép chia $(a^2-1)/8$, $(b^2-1)/8$, $(a-1)(b-1)/4$, và phép nâng (-1) lên luỹ thừa đó, ta không cần làm các phép chia cũng như nâng lên luỹ thừa, vì đòi hỏi quá nhiều phép tính bit. Vì giá trị luỹ thừa của (-1) chỉ phụ thuộc vào tính chẵn lẻ của các đại lượng trên, nên chẳng hạn đối với $(-1)^{(a^2-1)/8}$, giá trị đó chỉ phụ thuộc a mod 8 và bằng một trong những số của dãy sau đây:

$$\{0,1,0,-1,0,-1,0,1\}.$$

Thuật toán tính căn bậc 2 modulo p.

Trong nhiều ứng dụng (chẳng hạn, xem Chương 7), ta cần phải tính căn bậc 2 modulo p, khi biết nó tồn tại. Tất nhiên, một trong các phương pháp để giải phương trình đồng dư $x^2 \equiv a(mod\ p)$, $(a,p)=1$ là kiểm tra tất cả các số từ 1 đến $p-1$. Tuy nhiên, khi làm việc với p lớn, phương pháp này không thể áp dụng được (thời gian đòi hỏi là $O(p)$).

Với những số nguyên tố dạng $p \equiv 3(mod\ 4)$, bài toán khá đơn giản. Ta có:

$$x \equiv a^{(p+1)/4}(mod\ p).$$

Thật vậy,

$$x \equiv a^{(p+1)/2} \equiv a.a^{(p-1)/2} \equiv a(mod\ p)\ .$$

Khi $p \not\equiv 3(mod\ 4)$, ta có $p \equiv 1(mod\ 8)$ hoặc $p \equiv 5(mod\ 8)$. Trong trường hợp $p \equiv 5(mod\ 8)$, lời giải cũng có thể tìm được không khó khăn. Thật vậy, ta có:

$$a^{(p-1)/2} \equiv 1(\text{mod p}),$$

do đó

$$a^{(p-1)/4} \equiv \pm 1(\text{mod p}).$$

Dễ kiểm tra được rằng, trong trường hợp đồng dư thoả mãn với dấu cộng, nghiệm phải tìm là

$$x=a^{(p+3)/8}(\text{mod p}).$$

Nếu đồng dư thoả mãn với dấu trừ, dùng định lí 4.8 ta có:

$$a^{(p-1)/2} \equiv -1(\text{mod p}).$$

Từ đó nghiệm phải tìm là:

$$x=2a.(4a)^{(p-5)/8}(\text{mod p}).$$

Như vậy chỉ còn phải xét trường hợp $p \equiv 1(mod\ 8)$. Cho đến nay, mới chỉ có một thuật toán (thuật toán Shoof sử dụng đường cong elliptic) với thời gian đa thức. Tuy nhiên, trong thực tế, thuật toán đó rất khó sử dụng. Sau đây chúng ta tìm hiểu thuật toán xác suất của Tonelli và Shanks.

Thuật toán Tonelli-Shanks chính là một mở rộng tự nhiên của các trường hợp riêng đã xét trên đây.

Ta luôn luôn viết $p-1=2^e.q$, với q lẻ.

Nếu ta tìm được phần tử z và số nguyên chẵn k sao cho

$$a^q z^k \equiv 1 (\text{mod } p)$$

thì nghiệm cần tìm sẽ được cho bởi

$$x=a^{(q+1)/2} z^{k/2}.$$

Ta sẽ tìm phần tử z dưới dạng $z=n^q$.

Ta chỉ ra rằng, phần tử z như vậy thoả mãn điều kiện đặt ra khi và chỉ khi n là một không thặng dư bình phương modulo p. Ta có:

$$(a^q)^{2^e} = a^{\phi(p-1)} \equiv 1 (\text{mod } p),$$

do đó a^q thuộc vào nhóm G các phần tử cấp là một ước số của 2^e. Như vậy, để tồn tại k, chỉ cần chọn n là phần tử sinh của nhóm G (khi đó, do a là một không thặng dư bình phương nên số mũ k phải là chẵn). Số nguyên n sẽ là một phần tử sinh của G khi và chỉ khi $n, n^2, n^4,...,n^{2^e}$ ($\equiv 1 (mod\ p)$) không đồng dư với nhau modulo p. Dễ thấy rằng, điều đó xảy ra khi và chỉ khi n là một không thặng dư bình phương modulo p.

Để xây dựng thuật toán, ta cần giải quyết hai vấn đề: Tìm phần tử z, và tìm số mũ k.

Phần thứ nhất được giải quyết bằng thuật toán xác suất. Ta chọn ngẫu nhiên một số n, và tính kí hiệu Legendre $\left[\frac{n}{p}\right]$. Khi đó, nếu $\left[\frac{n}{p}\right]=-1$, ta đặt $z=n^q$. Trong trường hợp ngược lại, ta tiếp tục làm như trên với một số ngẫu nhiên khác cho đến khi tìm được một số n thích hợp. Vì số các thặng dư bình phương bằng $(p-1)/2$ nên mỗi lần chọn ngẫu nhiên một số n, xác suất để có $\left[\frac{n}{p}\right]=-1$ là 1/2.

Trong thực tế, ta có thể tìm ra một không thặng dư bình phương rất nhanh. Chẳng hạn, xác suất hai mươi lần thất bại liên tiếp nhỏ hơn 10^{-6}.

Số mũ k khó tìm hơn. Thật ra, ta không cần biết số mũ k, mà cần biết $a^{(q+1)/2} z^{k/2}$.

Thuật toán. Giả sử p là một số nguyên tố lẻ, $n \in Z$. Ta viết $p-1=2^e.q$ với q lẻ.

1. (Tìm phần tử sinh). Chọn ngẫu nhiên số n cho đến khi thoả mãn $\left[\frac{n}{p}\right]=-1$. Sau đó đặt $z \leftarrow n^q (mod\ p)$.

2. (Xuất phát). Đặt $y \leftarrow z$, $r \leftarrow e$, $x \leftarrow a^{(p-1)/2} (mod\ p)$, $b \leftarrow ax^2 (mod\ p)$, $x \leftarrow ax (mod\ p)$.

3. (Tìm số mũ). Nếu $b \equiv 1 (mod\ p)$ in ra x và kết thúc thuật toán, Trong trường hợp ngược lại, tìm số m nhỏ nhất sao cho $m \geq 1$, $b^{2^m} \equiv 1 (mod\ p)$. Nếu $m=r$, in ra thông

báo nói rằng a không phải là thặng dư bình phương modulo p.

4. (Thu hẹp số mũ). Đặt $t \leftarrow y^{2^{r-m-1}}$, $y \leftarrow t^2$, $r \leftarrow m$, $x \leftarrow xt$, $b \leftarrow by$ (mọi phép tính đều modulo p)và chuyển sang bước 3.

Chú ý rằng từ khi bắt đầu bước 3, ta luôn luôn có các đồng dư modulo p:

$$ax \equiv x^2,\ y^{2^{r-1}} \equiv -1,\ b^{2^{r-1}} \equiv 1.$$

Từ đó suy ra rằng, nếu nhóm con G_r các phần tử cấp là một ước của 2^r, thì y là phần tử sinh của nhóm G_r, $b \in G_{r-1}$, tức là b chính phương trong G_r. Vì r thực sự giảm tại mỗi bước lặp của thuật toán, nên số bước lặp nhiều nhất bằng e. Khi $r \leq 1$, ta có $b=1$, thuật toán kết thúc, và x là một căn bậc 2 của a mod p.

Có thể thấy rằng, trung bình, bước 3 và bước 4 đòi hỏi $e^2/4$ phép nhân modulo p, và nhiều nhất là e^2 phép nhân. Như vậy, thời gian chạy thuật toán là $O(\log^4 p)$.

§5. Số giả nguyên tố Euler.

Giả sử p là số nguyên tố lẻ và b là số nguyên không chia hết cho p. Khi đó theo tiêu chuẩn Euler ta có:

$$b^{(p-1)/2} \equiv \left[\frac{b}{p}\right] (\text{mod } p).$$

Như vậy, để kiểm tra một số n có phải là nguyên tố hay không, ta có thể lấy một số b nguyên tố cùng nhau với n, và kiểm tra xem đồng dư sau đây có đúng hay không:

$$b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n),$$

trong đó, vế bên phải là kí hiệu Jacobi. Nếu đồng dư thức đó không đúng thì n phải là hợp số. Nếu đồng dư thức trên đây nghiệm đúng, vẫn chưa kết luận được n có phải là nguyên tố hay không, nhưng "có nhiều khả năng" n là số nguyên tố.

Định nghĩa 4.18. Số nguyên dương n được gọi là *số giả nguyên tố Euler cơ sở b* nếu nó là một hợp số và đồng dư thức sau đây nghiệm đúng:

$$b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n)$$

Ta có mối liên hệ giữa số giả nguyên tố Euler cơ sở b và số giả nguyên tố cơ sở b đã xét trong chương 2.

Định lí 4.19. *Mọi số giả nguyên tố Euler cơ sở b đều là số giả nguyên tố cơ sở b.*

Chứng minh. Chỉ cần bình phương hai vế của đồng dư thức thoả mãn bởi các số giả nguyên tố Euler.

Điều ngược lại không đúng. Chẳng hạn, có thể thấy rằng số 431 là số giả nguyên tố cơ sở 2, nhưng không là số giả nguyên tố Euler cơ sở 2.

Định lí 4.20. *Mọi số giả nguyên tố mạnh cơ sở b đều là số giả nguyên tố Euler cơ sở b.*

Chứng minh. Cho n là số giả nguyên tố mạnh cơ sở b. Khi đó, nếu $n-1=2^s t$, trong đó t lẻ, thì, hoặc $b^t \equiv 1 (mod\ n)$, hoặc $b^{2^r t} \equiv -1 (mod\ n)$, với r nào đó $0 \le r \le s-1$. Giả sử $\prod_{j=1}^{m} p_j^{a_j}$ là phân tích của n thành thừa số nguyên tố. Ta xét riêng hai trường hợp.

Thứ nhất, $b^t \equiv 1 (mod\ n)$. Giả sử p là một ước nguyên tố của n. Khi đó $ord_p b | t$, và do đó $ord_p b$ là số lẻ. Mặt khác, $ord_p b$ là ước của $\phi(p)=p-1$, nên nó phải là ước của $(p-1)/2$. Vậy ta có

$$b^{(p-1)/2} \equiv 1 (mod\ p)$$

Theo tiêu chuẩn Euler, $\left[\frac{b}{p}\right]=1$, và do đó, $\left[\frac{b}{n}\right]=1$. Mặt khác ta có: $b^{(n-1)/2}=(b^t)^{2^s-1} \equiv 1 (mod\ p)$. Vậy n là số giả nguyên tố Euler cơ sở b.

Trường hợp thứ hai: $b^{2^r t} \equiv -1 (mod\ n)$. Nếu p là một ước nguyên tố của n thì $b^{2^r t} \equiv -1 (mod\ n)$.

Bình phương cả hai vế của dồng dư thức này ta được

$$b^{2^{r+1} t} \equiv 1 (\text{mod p}).$$

Từ đó suy ra $ord_p b | 2^{r+1} t$, nhưng $ord_p b$ không là ước của $2^r t$. Như vậy, $ord_p b = 2^{r+1} c$, trong đó c là một số nguyên lẻ. Mặt khác, vì $ord_p b | (p-1)$, $2^{r+1} | ord_p b$, nên $2^{r+1} | (p-1)$.

Như vậy, ta có: $p=2^{r+1} d+1$, trong đó d là số nguyên. Vì

$$b^{(ord_p b)/2} \equiv -1 (\text{mod p})$$

nên ta có:

$$\left[\frac{b}{p}\right] \equiv b^{(p-1)/2} = b^{(ord_p b/2)((p-1)/ord_p b)} \equiv (-1)^{(p-1)/ord_p b} = (-1)^{(p-1)/2^{r+1} c} (\text{mod p})$$

Vì c lẻ nên từ đó suy ra $\left[\frac{b}{p}\right]=(-1)^d$.

Bây giờ giả sử n có phân tích thành thừa số nguyên tố dạng:

$$n=\prod_{j=1}^{m} p_j^{\,a_j} .$$

Theo chứng minh phần trên, các ước nguyên tố p_i có dạng $p_i=2^{r+1}d_i+1$, và ta có:

$$\left[\frac{b}{n}\right]=\prod_{i=1}^{m}\left[\frac{b}{p_i}\right]_i^a=(-1)^{\sum_{i=1}^{m}a_i d_i} .$$

Mặt khác, dễ thấy rằng,

$$n\equiv 1+2^{r+1}\sum_{i=1}^{m}a_i d_i \pmod{2^{2r+2}}.$$

Do đó

$$t2^{s-1}=(n-1)/2\equiv 2^{r}\sum_{i=1}^{m}a_i d_i \pmod{2^{r+1}},$$

tức là

$$t2^{s-1-r}\equiv \sum_{i=1}^{m}a_i d_i \pmod 2$$

và

$$b^{(n-1)/2}=(b^{2^r t})^{2^{s-1-r}}\equiv(-1)^{2^{s-1-r}}=(-1)^{\sum_{i=1}^{m}a_i d_i} \pmod n$$

Như vậy,

$$b^{(n-1)/2}\equiv\left[\frac{b}{n}\right]\pmod n,$$

và n là số giả nguyên tố Euler cơ sở b.

Chú ý rằng, điều ngược lại không phải luôn luôn đúng: tồn tại những số giả nguyên tố Euler cơ sở b không là giả nguyên tố mạnh cơ sở đó. Ví dụ n=1105, b=2.

Tuy nhiên, với những điều kiện bổ sung, một số giả nguyên tố Euler sẽ là giả nguyên tố mạnh cùng cơ sở. Ta có định lí sau.

Định lí 4. 21. *Số n giả nguyên tố Euler cơ sở b là số giả nguyên tố mạnh cơ sở b nếu* $n\equiv 3 \pmod 4$, *hoặc* $\left[\frac{b}{n}\right]=-1$.

Chứng minh. Trường hợp thứ nhất: $n\equiv 3 \pmod 4$. Khi đó $n-1=2.t$ và t lẻ. Vì n là số giả nguyên tố Euler cơ sở b nên

$$b^t = b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n).$$

Như vậy, n là số giả nguyên tố mạnh cơ sở b.

Trong trường hợp thứ hai, ta viết $n-1=2^s t$, trong đó t lẻ, s là số nguyên dương. Vì n là số giả nguyên tố mạnh cơ sở b nên

$$b^{2^{s-1}t} = b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n).$$

Theo giả thiết ta có:

$$b^{t2^{s-1}} \equiv -1 (\text{mod } n).$$

Như vậy n là số giả nguyên tố mạnh cơ sở b.

Dùng số giả nguyên tố Euler, ta có thể xây dựng thuật toán xác suất để kiểm tra một số là nguyên tố hay không. Thuật toán này được Solovay và Strassen tìm ra đầu tiên năm 1977 ([S-S]).

Ta bắt đầu bằng bổ đề sau.

Bổ đề 4.22. *Giả sử n là một số nguyên dương lẻ không chính phương. Khi đó tồn tại ít nhất một số b với 1<b<n, (b,n)=1, sao cho* $\left[\frac{b}{n}\right] = -1$.

Chứng minh. Nếu n là nguyên tố, số b tồn tại theo định lí 4.3. Khi n là hợp số không chính phương, ta viết $n=rs$, trong đó $(r,s)=1$ và $r=p^e$, với p là một số nguyên tố lẻ và e số nguyên dương lẻ. Bây giờ giả sử t là một không thặng dư bình phương của số nguyên tố p. Ta dùng định lí Trung Quốc về phần dư để tìm số nguyên b sao cho $1<b<n$, $(b,n)=1$ và

$$b \equiv t (\text{mod } r)$$

$$b \equiv 1 (\text{mod } s)$$

Khi đó ta có $\left[\frac{b}{r}\right] = \left[\frac{b}{p^e}\right] = (-1)^e = -1, \left[\frac{b}{s}\right] = 1$, tức là $\left[\frac{b}{n}\right] = -1$.

Bổ đề 4.23. *Với mỗi hợp số lẻ n, tồn tại ít nhất một số b sao cho 1<b<n, (b,n)=1 và*

$$b^{(n-1)/2} \not\equiv \left[\frac{b}{n}\right] (mod\ n). \qquad (3.1)$$

Giả sử ngược lại, với mọi số nguyên không vượt quá n và nguyên tố cùng nhau với n, ta có

$$b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n).$$

Từ đó suy ra, nếu *(b,n)=1* thì

$$b^{(n-1)} \equiv 1(\text{mod } n).$$

Như vậy, *n* phải là số Carmicheal, và do đó, $n=q_1q_2...q_r$ là tích của các số nguyên tố lẻ khác nhau. Ta sẽ chỉ ra rằng

$$b^{(n-1)/2} \equiv 1(\text{mod } n).$$

đối với mọi số nguyên *b* không vượt quá *n* và nguyên tố cùng nhau với *n*.

Giả sử ngược lại, tồn tại *b* thoả mãn

$$b^{(n-1)/2} \equiv -1(\text{mod } n).$$

Dùng định lí Trung Quốc về phần dư, ta tìm được số *a, 1<a<n, (a,n)=1* sao cho

$$a \equiv b(\text{mod } q_1)$$

$$a \equiv 1(\text{mod } q_2q_3...q_r)$$

Như vậy

$$a^{(n-1)/2} \equiv b^{(n-1)/2} \equiv -1(\text{mod } q_1)$$

$$a^{(n-1)/2} \equiv 1(\text{mod } q_2q_3...q_r)$$

Do đó

$$a^{(n-1)/2} \not\equiv \pm 1(\text{mod } n),$$

trái với giả thiết phản chứng (3.1).

Như vậy, với mọi *b, 1<b<n, (b,n)=1* ta có:

$$b^{(n-1)/2} \equiv 1(\text{mod } n).$$

Từ đồng dư trên và (3.1) ta có:

$$b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] \equiv 1(\text{mod } n),$$

mâu thuẫn với bổ đề 4.22. Bổ đề 4.23 được chứng minh.

Định lí trên đây được dùng làm cơ sở cho một thuật toán kiểm tra nguyên tố xác suất. Ta có định lí sau.

Định lí 4.24. *Đối với mỗi hợp số lẻ n, tồn tại không quá $\phi(n)/2$ số nguyên dương b nhỏ hơn n, nguyên tố cùng nhau với n, sao cho n là số giả nguyên tố mạnh Euler cơ sở b.*

Chứng minh. Theo bổ đề 4.23., tồn tại số *b, 1<b<n, (b,n)=1* sao cho

$$b^{(n-1)/2} \not\equiv \left[\frac{b}{n}\right] (\text{mod } n).$$

Giả sử $a_1,a_2,...,a_m$ là các số thoả mãn $1 \le a_j<n$, $(a_j,n)=1$ và

$$a_j^{(n-1)/2} \equiv \left[\frac{a_j}{n}\right] = 1 (\text{mod } n)$$

Giả sử $r_1,r_2,...,r_m$ là thặng dư dương bé nhất của các số $ba_1,ba_2,...,ba_m$. Các số r_j khác nhau và nguyên tố cùng nhau với n. Ta sẽ chứng tỏ rằng chúng không thoả mãn đồng dư thức như đối với các số a_j. Thật vậy, nếu ngược lại

$$r_j^{(n-1)/2} \equiv \left[\frac{r_j}{n}\right] = 1 (\text{mod } n)$$

thì ta có:

$$ba_j^{(n-1)/2} \equiv \left[\frac{ba_j}{n}\right] = 1 (\text{mod } n)$$

và như vậy:

$$b^{(n-1)/2} a_j^{(n-1)/2} \equiv \left[\frac{b}{n}\right]\left[\frac{a_j}{n}\right]$$

Từ đó suy ra:

$$b^{(n-1)/2} \equiv \left[\frac{b}{n}\right] (\text{mod } n),$$

mâu thuẫn với tính chất của b.

Như vậy, tập hợp các số a_j và r_j không giao nhau. Gộp cả hai tập hợp này, ta được $2m$ số khác nhau, bé hơn n và nguyên tố cùng nhau với n. Từ đó suy ra $m<\phi(n)/2$, định lí được chứng minh.

Nhận xét. Từ định lí trên, ta thấy rằng, nếu n là một hợp số lẻ, b là số chọn ngẫu nhiên trong các số $1,2,...,n-1$, thì xác suất để n là giả nguyên tố Euler có sở b sẽ bé hơn 1/2. Ta có định lí sau.

Định lí 4.25. (Thuật toán kiểm tra nguyên tố xác suất Solovay-Strassen). *Cho n là một số nguyên dương. Ta chọn ngẫu nhiên k số $b_1,b_2,...,b_k$ từ các số $1,2,...,n-1$. Đối với mỗi số nguyên b_j, xét đồng dư thức*

$$b_j^{(n-1)/2} \equiv \left[\frac{b_j}{n}\right] (\mathrm{mod}\, n)$$

-Nếu một trong các đồng dư thức đó không nghiệm đúng thì n là hợp số.

-Nếu n là nguyên tố thì mọi đồng dư thức đều nghiệm đúng.

-Nếu n là hợp số, thì xác suất để mọi đồng dư thức nghiệm đúng là bé hơn $1/2^k$.

Như vậy, nếu k đủ lớn, và n trải qua được kiểm tra xác suất trên đây, thì "hầu như chắc chắn" n là số nguyên tố.

Nhận xét. 1) Vì mọi số giả nguyên tố mạnh cơ sở *b* đều là số giả nguyên tố Euler cơ sở *b*, nên số các hợp số *n* trải qua được kiểm tra xác suất Solovay-Strassen lớn hơn số các hợp số trải qua được kiểm tra Rabin. Cả hai thuật toán kiểm tra này đều cần $O(k(log_2 n)^3)$ phép tính bit.

2) Chẳng hạn, nếu *n* là số trải qua kiểm tra xác suất Solovay-Strassen với $k=40$. Khi đó *n* là hợp số với xác suất nhỏ hơn 2^{-40} tương đương 10^{-12}, bé hơn xác suất để phần cứng máy tính mắc một sai lầm!

Bài tập và tính toán thực hành chương 4

I. Bài tập

4.1. Tìm tất cả các số tự nhiên *b* sao cho 15 và 21 là các số giả nguyên tố cơ sở b.

4.2. Chứng minh rằng tồn tại 36 cơ sở *b* (modulo 91) để 91 là số giả nguyên tố cơ sở *b*.

4.3. Giả sử *p* và *2p-1* đều là số nguyên tố. Chứng minh rằng *n=p(2p-1)* là số giả nguyên tố đối với một nửa số cơ sở *b*.

4.4. Chứng minh rằng tồn tại vô hạn số nguyên tố dạng *4k+1*.

4.5. Chứng minh rằng tồn tại vô hạn số nguyên tố có các dạng sau đây: a) *8k+3, b)8k+5, c) 8k+7*.

4.6. Chứng minh rằng nếu *p* là một số nguyên tố dạng *4k+3* và *q=2p+1* cũng là số nguyên tố, thì $q|M_p=2^p-1$.

4.7. Chứng minh rằng $23|M_{11}$, $47|M_{23}$,$503|M_{251}$.

4.8. Chứng minh rằng 1105 là số giả nguyên tố Euler cơ sở 2 và không giả nguyên tố mạnh cơ sở 2.

4.9. Chứng minh rằng 15841 là: a) số giả nguyên tố mạnh cơ sở 2; b) số giả nguyên tố Euler cơ sở 2; c) số Carmichael.

4.10. Chứng minh rằng nếu *n* là số giả nguyên tố mạnh Euler cơ sở *a* và *b* thì *n* cũng là số giả nguyên tố mạnh Euler cơ sở *ab*.

4.11. Chứng minh rằng nếu *n* là số giả nguyên tố Euler cơ sở 2, và nếu $n \equiv 5 (mod\ 8)$ thì *n* là số giả nguyên tố mạnh cơ sở 2.

4.12. Chứng minh rằng nếu *n* là số giả nguyên tố Euler cơ sở *b* thì *n* cũng là số giả nguyên tố Euler cơ sở *n-b*.

4.13. Chứng minh rằng nếu *n* là số giả nguyên tố Euler cơ sở 3 và $n \equiv 5 (mod\ 12)$ thì *n* là số giả nguyên tố mạnh cơ sở 3.

II. Thực hành trên máy tính

II. 1. Thực hành kiểm tra một số là thặng dư bình phương

Cho *a, b* là các số nguyên. để kiểm tra xem *a* có phải là thặng dư bình phương của *b* hay không ta thực hiện dòng lệnh như sau:

```
[>quadres(a, b);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình hiện lên số 1 thì *a* là thặng dư bình phương của *b*, nếu trên màn hình hiện lên số -1 thì không phải.

Thí dụ: 74 có phải là thặng dư bình phương của 101 hay không?

Ta thực hiện lệnh

```
[>quadres(74,101);
```

```
                                -1
```

74 không phải là thặng dư bình phương của 101

II. 2. Thực hành tính ký hiệu Legendre

Cho *a* là số nguyên, *p* là số nguyên tố. Để tính kí hiệu Legendre của *a* và *p* ta thực hiện lệnh như sau:

```
[legendre(a,p);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Tính $\begin{bmatrix} 9 \\ 11 \end{bmatrix}$.

Ta thực hiện lệnh

```
[legendre(9,11);
```

```
                                1
```

Chú ý: Khi thực hiện lệnh tính lí hiệu Legendre, máy tính sẽ cho kết quả là 0, 1, hoặc -1. Nếu kết quả là 0 thì *a* chia hết cho *p*. Nếu kết quả là 1 thì *a* là thặng dư bình phương của *p*. Nếu kết quả là -1 thì *a* không là thặng dư của *p*. Do đó ta cũng có thể dùng dòng lệnh trên để kiểm tra thặng dư bình phương.

II. 3. Tính kí hiệu Jacobi

Cho *b* là số nguyên dương lẻ, *a* nguyên tố cùng nhau với *b*. Để tính kí hiệu Jacobi của *a* và *b* ta thực hiện dòng lệnh như sau:

```
[jacobi(a, b);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả.

Thí dụ: Tính $\begin{bmatrix} 26 \\ 35 \end{bmatrix}$

Ta thực hiện lệnh:

```
[> jacobi(26,35);
```

```
                                -1
```

Thí dụ: Tính $\begin{bmatrix} 28 \\ 21 \end{bmatrix}$

Ta thực hiện lệnh:

```
[> jacobi(28,21);
                                    0
```

Nếu kết quả là 0 thì a và b không nguyên tố cùng nhau.

II. 4. Tìm căn bậc 2 modulo một số

Cho x, n là các số nguyên. Để tìm căn bậc 2 của x modulo n ta thực hiện dòng lệnh như sau:

```
[>msqrt(x,n);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả. Nếu căn không tồn tại trên màn hình sẽ xuất hiện chữ "FAIL".

Thí dụ: Tính căn bậc 2 của 3 modulo 11.

Ta thực hiện như sau:

```
[>msqrt(3,11);
                                    5
```

Thí dụ: Tính căn bậc 2 của 3 modulo 7.

Ta thực hiện như sau:

```
[>msqrt(3,7);
                                  FAIl
```

II. 5. Thực hành kiểm tra số giả nguyên tố Euler

Để kiểm tra số nguyên dương n cho trước có phải là số giả nguyên tố Euler cơ sở b hay không ta thực hiện theo các bước sau:

Bước 1: Kiểm tra tính nguyên tố của n, ta thực hiện bằng dònglệnh

```
[>isprime(n);
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình xuất hiện chữ "true" thì n là số nguyên tố, khi đó ta khẳng định n không phải là số giả nguyên tố Euler cơ sở b. Nếu trên màn hình xuất hiện chữ "false" ta tiếp tục thực hiện bước 2.

Bước 2: Tính kí hiệu Jacobi $J:=\left[\frac{b}{n}\right]$ của n và b, thực hiện bằng dòng lệnh

```
[> J:= jacobi(b,n);
```

Sau dấu (;) ấn phím "Enter".

Bước 3: Kiểm tra đồng dư thức $b^{(n-1)/2} \equiv J(\text{mod } n)$, thực hiện bằng dòng lệnh

```
[>b^((n-1)/2)-J mod n;
```

Sau dấu (;) ấn phím "Enter". Nếu trên màn hình xuất hiện số 0 thì *n* là số giả nguyên tố Euler cơ sở *b*.

Thí dụ: Số 1105 có phải là số giả nguyên tố Euler cơ sở 2 hay không?

Ta thực hiện lệnh như sau:

```
[> isprime(1105);
                                  false
[> J:=J(1105,2);
                                    1
[> 2^((1105-1)/2)-J mod 1105;
                                    0
```

Vậy 1105 là số giả nguyên tố Euler cơ sở 2.

Chương 5

TRƯỜNG VÀ ĐA THỨC

§1. Định nghĩa.

Một trong những khái niệm cơ bản của đạI số và số học là các trường hữu hạn, Trong chương này, Chúng tôi sẽ trình bày những kiến thức cơ bản nhất về các trường hữu hạn, cần thiết khi tìm hiểu những ứng dụng mới của số học. Ngoài ra, chúng tôi cố gắng minh hoạ một trong những "động lực" của sự phát triển số học hiện đại: sự tương tự giữa số và đa thức. Một vài kết quả gần đây về sự tương tự đó sẽ được đề cập tới.

Để tiện lợi cho bạn đọc khi sử dụng, chúng tôi nhắc lại ở đây những khái niệm cần thiết.

Trường là một tập hợp K có quá một phần tử, được trang bị hai phép tính cộng và nhân thoả mãn các quy tắc sau đây (các chữ cái la tinh chỉ các phần tử tuỳ ý của trường)

1. (tính chất giao hoán của phép cộng): $a+b=b+a$
2. (tính chất kết hợp của phép cộng): $a+(b+c)=(a+b)+c$
3. (tồn tại phần tử 0): Tồn tại $0 \in K$ sao cho $0+a=a+0=a$
4. (tồn tại $-a$): Tồn tại $-a \in K$ sao cho $a+(-a)=0$
5. (tính chất giao hoán của phép nhân): $ab=ba$
6. (tính chất kết hợp của phép nhân): $a(bc)=(ab)c$
7. (tồn tại đơn vị): Tồn tại $1 \in K$ sao cho $1a=a$
8. (tồn tại a^{-1}): Tồn tại $a^{-1} \in K$ sao cho $aa^{-1}=1$
9. (luật phân bố của phép nhân đối với phép cộng): $a(b+c)=ab+ac$

Những ví dụ thường gặp nhất là: trường Q các số hữu tỷ, trường R các số thực, trường C các số phức. Các trường đó đều có vô hạn phần tử.

Trong nhiều vấn đề lí thuyết cũng như ứng dụng, ta thường làm việc với các trường chỉ có hữu hạn phần tử. Chẳng hạn, có thể thấy rằng, các thặng dư không âm bé nhất modulo p, lập thành một trường có p phần tử. Sau đây, ta sẽ thấy rằng, đó chính là trường cơ bản để xây dựng nên tất cả các trường hữu hạn.

Giả sử p là số nguyên tố. Kí hiệu qua F_p trường có p phần tử. Rõ ràng khi cộng p lần phần tử 1 của trường, ta được 0. Do đó, $pa=0$ với mọi phần tử $a \in F_p$. Với một trường

K tuỳ ý, số *p* không âm bé nhất sao cho *p1=0* được gọi là *đặc trưng* của trường *K*. Chẳng hạn, *Q, R, C* là các trường có đặc trưng *0*, F_p là trường có đặc trưng bằng *p*. Dễ thấy rằng, mọi trường hữu hạn đều có đặc trưng khác không, và đặc trưng của nó là một số nguyên tố.

§2. Mở rộng trường.

Giả sử $k \subset K$ là các trường, đồng thời các phép cộng và nhân trong *k* chính là cảm sinh bởi các phép tính tương ứng trong *K*. Khi đó, *K* được gọi là *mở rộng* của trường *k*.

Ví dụ. *C* là mở rộng của *R, R* là mở rộng của *Q*.

Nếu tồn tại các phần tử $\alpha_1, \alpha_2,..., \alpha_n \in K$ sao cho mọi phần tử của $a \in K$ đều có thể biểu diễn dưới dạng

$$a=k_1\alpha_1+k_2\alpha_2+...+k_n\alpha_n$$

trong đó $k_1,k_2,...,k_n$ là các phần tử của trường *k*, thì *K* được gọi là *mở rộng hữu hạn* của *k*. Trong trường hợp này, *K* là một *không gian vectơ hữu hạn chiều trên k*. Ta nói *K* là *mở rộng hữu hạn của k sinh bởi* $\alpha_1, \alpha_2,..., \alpha_n$.

Ví dụ. *C* là mở rộng của trường *R*, sinh bởi phần tử *i*, nói cách khác, sinh bởi nghiệm cuả phương trình $x^2+1=0$.

Ta nói *C* là *trường nâng của R bởi đa thức* $P(x)=x^2+1$. Sau đây, ta sẽ thấy rằng, mọi mở rộng hữu hạn của các trường đều được thực hiện bằng cách tương tự như trên.

Định nghĩa 5.1. Giả sử *K* là một mở rộng của *k*. Phần tử $a \in K$ được gọi là *đại số* trên trường *k* nếu nó là nghiệm của một đa thức với hệ số trên trường *k*.

Nếu thêm điều kiện hệ số của luỹ thừa cao nhất bằng 1 thì đa thức xác định duy nhất đối với mỗi phần tử đại số trên *k*. *K* được gọi là *trường nâng* của *k* bởi đa thức *P(x)* nếu nó là mở rộng của *k* bởi các nghiệm của đa thức *P(x)*.

Đối với các đa thức, ta cũng có các tính chất hoàn toàn tương tự như đối với các số nguyên.

Đối với một trường *k* tuỳ ý, ta kí hiệu qua *k[x]* vành các đa thức với hệ số trong *k*. Đa thức *P* được gọi là *chia hết* cho đa thức *Q* nếu tồn tại đa thức *R* sao cho *P=QR*. Một đa thức không chia hết cho đa thức nào bậc nhỏ hơn, khác hằng số được gọi là *đa thức bất khả quy*. Hai đa thức không có ước chung nào khác hằng được gọi là *nguyên tố cùng nhau*. Một đa thức trong *k[x]* luôn luôn phân tích được thành tích của các đa thức bất khả quy. Phân tích đó là duy nhất, nếu đòi hỏi các đa thức ban đầu cũng như đa thức trong khai triển đều có hệ số của luỹ thừa cao nhất bằng 1.

Đối với mỗi đa thức *P(x)* trên trường *k*, bao giờ cũng tồn tại một trường *K* mở rộng của *k* sao cho *P(x)* phân tích được thành các đa thức bậc nhất trên *K*. Như vậy, nếu hệ số của luỹ thừa cao nhất trong *P(x)* là 1 thì *P(x)* được phân tích dưới dạng:

$$P(x)=(x-\alpha_1)(x-\alpha_2)...(x-\alpha_n), \quad \text{với } \alpha_i \in K, i=1,...,n.$$

Nếu đối với đa thức hệ số trong *k,* phân tích trên đây có được với $\alpha_i \in k$, trường *k* được gọi là *đóng đại số*. Nói cách khác, trường đóng đại số là trường chứa mọi nghiệm của các đa thức với hệ số trong trường đó. Như vậy, trong trường đóng đại số, các đa thức bất khả quy chỉ có thể là đa thức bậc nhất. Chẳng hạn trường số phức *C* là trường đóng đại số, trường các số thực *R* không đóng đại số. Thương của hai đa thức với hệ số trong *k* được gọi là *hàm hữu tỷ* trên *k*.

§3. Trường hữu hạn.

Như đã nói, trường gồm hữu hạn phần tử có đặc trưng khác không, và đặc trưng đó là một số nguyên tố *p*.

Giả sử F_q là một trường hữu hạn gồm *q* phần tử, đặc trưng *p*. Vì F_q chứa phần tử 1 nên nó sẽ chứa trường F_p như một trường con. Do F_q là trường hữu hạn nên nó là mở rộng hữu hạn của F_p, nghĩa là một không gian vectơ *r* chiều trên F_p. Từ đó suy ra rằng F_q gồm p^r phần tử, tức là $q=p^r$.

Ngược lại, ta sẽ chứng tỏ rằng, với *p, r* cho trước (*p* là số nguyên tố và *r* là số nguyên dương), tồn tại trường với p^r phần tử. Hơn nữa, các trường hữu hạn với số phần tử như nhau sẽ đẳng cấu với nhau, nghĩa là có tương ứng 1-1 giữa chúng, và tương ứng này bảo toàn các phép tính cộng và nhân, phần tử 0 và phần tử nghịch đảo của trường.

Ta có định lí sau.

Định lí 5.2. *Giả sử F_q là trường hữu hạn với $q=p^r$ phần tử. Khi đó, mọi phần tử của F_q đều thoả mãn phương trình*

$$X^q-X=0,$$

và F_q chính là tập hợp các nghiệm của phương trình đó. Ngược lại, trường nâng của F_p bởi đa thức X^q-X là trường hữu hạn có q phần tử.

Chứng minh định lí này hoàn toàn tương tự như chứng minh định lí 3.27 Chương 3, và được dành cho độc giả .

Giả sử F_q là trường có *q* phần tử. Ta kí hiệu qua F_q^* tập hợp các phần tử khác không của trường F_q. Khi đó, mọi phần tử của F_q^* đều có nghịch đảo, và F_q^* lập thành một nhóm Aben. Vì F_q^* có hữu hạn phần tử, nên đối với một phần tử tuỳ ý $a \in F_q^*$, tồn tại số nguyên không âm *k* sao cho $a^k=1$. Số *k* bé nhất thoả mãn tính chất đó được gọi là bậc của phần tử *a*.

Đối với mọi phần tử *a* tuỳ ý, bậc của *a* luôn là một ước của *q-1*. Chứng minh điều này cũng hoàn toàn tương tự như khi chứng minh bậc của một số modulo *n* là ước của $\phi(n)$ (xem hệ quả 3.20 Chương 3).

Giả sử g là một phần tử của F_q^*, và bậc của g đúng bằng $q-1$. Khi đó, tập hợp $\{g,g^2,...,g^{q-1}\}$ chính là tất cả các phần tử của F_q^*. Ta nói g là phần tử sinh của nhóm F_q^*.

Định lí sau đây là một tương tự của định lí 3.26 trong chương 3.

Định lí 5.3. *Mọi trường hữu hạn đều có phần tử sinh. Nếu g là một phần tử sinh của F_q^* thì g^s là phần tử sinh của F_q^* khi và chỉ khi $(s,q-1)=1$. Như vậy, tồn tại tất cả $\phi(q-1)$ phần tử sinh của F_q^*.*

Bây giờ ta sẽ mô tả cụ thể cách xây dựng trường F_q từ trường F_p.

Để dễ hình dung, trước tiên ta xét việc xây dựng trường số phức C như là một trường nâng của số thực R bởi đa thức $P(x)=x^2+1$. Như ta đã biết, có thể xem mỗi số phức như một cặp số thực (a,b), và do đó, có thể đồng nhất mỗi số phức với một đa thức $ax+b$ hệ số thực. Với cách tương ứng như vậy, khi nhân hai số phức (biểu diễn bởi hai đa thức), ta chỉ việc nhân theo quy tắc nhân các đa thức, và thay x^2 bởi (-1). Nói cách khác, tập hợp các số phức chính là tập hợp các đa thức với hệ số thực, trong đó hai đa thức được đồng nhất khi và chỉ khi hiệu của chúng bằng đa thức $P(x)=x^2+1$. Ta viết $C=R[x]/P(x)$.

Trường F_q, $q=p^r$, được xây dựng từ trường F_p theo cách hoàn toàn tương tự. Ta xuất phát từ một đa thức bất khả quy $P(x)$ bậc r với hệ số trong F_p, trong đó hệ số của x^r bằng 1. Khi đó ta có:

$$F_q=F_p[x]/P(x).$$

Như vậy, các phần tử của F_q là các đa thức với hệ số trong F_p, bậc bé hơn r (vì giả sử $P(x)=x^r+a_{r-1}x^{r-1}+...+a_0$, khi đó x^r sẽ được thay bởi $-(a_{r-1}x^{r-1}+...+a_0)$).

Ta có thể xuất phát từ đa thức bất khả quy tuỳ ý. Các trường nhận được có số phần tử như nhau , và đẳng cấu với nhau.

Ta minh hoạ những điều nói trên qua ví dụ cụ thể.

Ví dụ. Xây dựng trường F_{16} từ trường F_2 bởi đa thức

$$P(x)=x^4+x^3+x^2+x+1.$$

Đa thức đang xét là một đa thức bất khả quy trên trường F_2. Thật vậy, nếu nó có ước khác hằng số thì phải có ước là đa thức bậc 1 hoặc bậc 2. Nếu ước là đa thức bậc 1, $P(x)$ có nghiệm trong F_2: điều này không xảy ra vì $P(0)=P(1)=1$. Có bốn đa thức bậc 2 trên F_2 đó là các đa thức x^2, x^2+1, x^2+x, x^2+x+1. Thử trực tiếp cho thấy rằng không có cặp đa thức nào có tích bằng $P(x)$.

Các phần tử của F_{16} là các đa thức bậc bé hơn hoặc bằng 3, với hệ số 0 hoặc 1:

-Bậc 0: $0,1$.

-Bậc 1: x, $x+1$.

-Bậc 2: x^2, x^2+1, x^2+x, x^2+x+1.

-Bậc 3: x^3, x^3+1, x^3+x, x^3+x^2, x^3+x+1, x^3+x_2+x+1, x_3+x^2+x, x^3+x^2+x+1.

Quy tắc cộng và nhân là quy tắc cộng và nhân thông thường của các đa thức, với chú ý $1+1=0$ và $x^4=-(x^3+x^2+x+1)$.

Trong nhiều ứng dụng, chẳng hạn trong lí thuyết thông tin, người ta thường viết các phần tử của trường F_q theo các hệ số của chúng. Như trong ví dụ trên đây, các phần tử của trường sẽ là: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1100, 1001, 1101, 1110, 1111, cùng với một bảng để cho quy tắc cộng và nhân của chúng. Chú ý rằng, các quy tắc này khác với các quy tắc số học đồng dư modulo q.

Khi biết một phần tử sinh của trường F_q, ta có thể tìm các phần tử khác bằng cách nâng lên luỹ thừa. Sau đây, ta sẽ tìm hiểu một thuật toán thời gian đa thức để làm việc đó. Thuật toán này sẽ được áp dụng trong những chương sau.

Trước khi đi vào mô tả thuật toán, để dễ hình dung, ta xét ví dụ sau đây. Giả sử ta cần tính $(1994)^{23}(\text{mod } 4611)$. Nếu dùng cách thông thường (tính lần lượt các luỹ thừa của 1994), ta phải làm 22 phép nhân và 22 phép chia. Để giảm bớt số phép tính phải làm, ta dùng phương pháp bình phương liên tiếp như sau. Ta có:

$$(1994)^{23}=(1994)^{16+4+2+1}$$

Như vậy, ta chỉ cần tính modulo của các luỹ thừa 1,2,4,8,16 của 1994. Nói cách khác, ta chỉ cần làm phép bình phương liên tiếp 4 lần, sau đó nhân các kết quả ở những luỹ thừa nào tương ứng với số 1 trong biểu diễn số 23 dưới dạng cơ số 2. Ta có $23=(10111)_2$, nên ta nhân kết quả của những luỹ thừa 16,4,2,1.

Cách làm như trên áp dụng được cho mọi nhóm nhân. Giả sử $g \in G$ là phần tử của nhóm nhân G nào đó, ta cần tính g^n, với n là số tự nhiên. Ta viết n dưới dạng cơ số 2: $n=\sum \varepsilon_i 2^i$, trong đó $\varepsilon = \pm 1$. Khi đó ta tính

$$g^n=\prod_{\varepsilon_i=1}(g^i).$$

Thuật toán.

S1. (Xuất phát) đặt $y \leftarrow 1$. Nếu $n=0$, thuật toán kết thúc. Nếu ngược lại, đặt $N \leftarrow n$, $z \leftarrow g$.

S2. (Nhân). Nếu N lẻ, đặt $y \leftarrow z.y$.

S3. (Chia đôi N). Đặt $N \leftarrow [N/2]$. Nếu $N=0$, in ra y và kết thúc thuật toán. Ngược lại, đặt $z \leftarrow z.z$ và quay về S2.

Có thể chứng minh tính đúng đắn của thuật toán với nhận xét rằng, từ bước thứ hai trở đi, ta luôn luôn có $g^n=y.z^N$.

Ta sẽ đánh giá độ phức tạp của thuật toán nói trên.

Số phép nhân phải làm bằng số chữ số của n, cộng thêm số chữ số 1 trong cách viết nhị phân của n, và trừ đi 1. Như vậy, số phép nhân không vượt quá $2[\log n]+1$, tức là $O(\log n)$. Nếu ta tính trong lớp đồng dư modulo m nào đó, mỗi phép nhân đòi hỏi $O(\log^2 m)$ phép tính bit, và toàn bộ số phép tính bit cần thiết sẽ là $O(\log n \log^2 m)$.

Như ta đã thấy ở trên, để thực hiện các phép tính trên trường F_q, ta phải làm các phép tính đối với các đa thức. Sau đây là vài thuật toán để thực hiện các phép tính đó.

Thuật toán chia

Xét các đa thức với hệ số trong trường K tuỳ ý. Với mỗi đa thức P, kí hiệu qua $l(P)$ hệ số của luỹ thừa cao nhất. Ta có thuật toán để, với đa thức đã cho A, B, $B \neq 0$, tìm các đa thức Q, R sao cho $A=BQ+R$, và $\deg R<\deg B$:

C1. (Xuất phát). Đặt $R \leftarrow A$, $Q \leftarrow 0$.

C2. (Kết thúc?). Nếu $\deg R<\deg B$, kết thúc thuật toán.

C3.(Tìm hệ số). Đặt $S \leftarrow \frac{l(R)}{l(B)} x^{\deg R-\deg B}$. Sau đó, đặt $Q \leftarrow Q+S$, $R \leftarrow R-S.B$, và chuyển sang bước C2.

Ta cần lưu ý ngay một điều. Về mặt lí thuyết, sau bước S3, bậc của R phải giảm (vì hệ số của $x^{\deg R}$ sẽ bằng 0 do định nghĩa của S). Tuy nhiên, khi làm việc trên máy, thực tế là ta chỉ có các số gần đúng, nên có thể xảy ra trường hợp hệ số của $x^{\deg R}$ tuy rất nhỏ, nhưng khác không, nghĩa là bậc không giảm, và do đó không bảo đảm là thuật toán kết thúc! Vì thế, khi viết chương trình, nhất thiết phải để hệ số đó bằng 0 sau phép tính $R \leftarrow R-S.B$.

Để tìm ước chung lớn nhất của các đa thức, ta có thuật toán Euclid sau đây.

Thuật toán

Cho các đa thức A,B, tìm ƯCLN của A,B.

EP1. (Kết thúc?) Nếu $B=0$, in ra A và kết thúc thuật toán.

EP2. (Bước Euclid). Giả sử $A=BQ+R$, với $\deg R<\deg B$ (tính bằng thuật toán C trình bày ở trên). Đặt $A \leftarrow B$, $B \leftarrow R$, và quay về bước EP1.

Định lí 5.4. *Có thể nhân hoặc chia hai phần tử của trường F_q với $O(\log^3 q)$ phép tính bit. Nếu k là số nguyên dương thì một phần tử của F_q có thể nâng lên luỹ thừa k với $O(\log k \log^3 q)$ phép tính bit.*

Chứng minh. Giả sử F_q được xây dựng bằng cách nâng trường F_p bởi đa thức bất khả quy $P(x)$ bậc r. Khi đó, các phần tử của trường F_q chính là các đa thức với hệ số trong trường F_p modulo đa thức $P(x)$. Để nhân hai phần tử của trường F_q, ta phải nhân hai đa thức như vậy. Để làm việc đó, ta phải thực hiện $O(r^2)$ phép nhân modulo p (vì các đa thức có bậc nhỏ hơn r), cùng với một số phép tính cộng. Các phép này đòi hỏi thời gian ít hơn. Sau khi có kết quả của phép nhân, ta lại phải tính "modulo đa thức P(x)", nghĩa là làm phép chia cho đa thức $P(x)$ để biết được phần dư. Phép chia đa thức này đòi hỏi $O(r)$ phép chia các số nguyên modulo p và $O(r^2)$ phép nhân các số nguyên modulo p Như ta đã biết, mỗi phép nhân số nguyên modulo p có thể thực hiện bằng $O(\log^2 p)$ phép tính bit , còn mỗi phép chia modulo p có thể làm (chẳng hạn, dùng thuật toán Euclid) với $O(\log^3 p)$ phép tính bit. Như vậy, toàn bộ số

phép tính bit được thực hiện khi nhân hai phần tử của trường F_q là: $O(r^2\log^2 p+r\log^3 p)=O((r\log p)^3)=O(\log^3 q)$. Khẳng định của định lí được chứng minh đối với phép nhân.

Xét phép chia các phần tử của F_q. Để chứng minh rằng có thể hiện phép chia sau $O(\log^3 q)$ phép tính bit, ta chỉ cần chứng tỏ rằng, nghịch đảo của một phần tử tìm được bởi $O(\log^3 q)$ phép tính bit, rồi áp dụng kết quả đã chứng minh đối với phép nhân.

Giả sử ta cần tìm nghịch đảo của phần tử $Q \in F_q$ (là một đa thức bậc nhỏ hơn r, hệ số trong F_p). Dùng thuật chia Euclid cho các đa thức trên trường F_q, ta cần biểu diễn 1 như là tổ hợp tuyến tính của đa thức $P(x)$ và $Q(x)$. Điều này làm được bởi $O(r)$ phép chia các đa thức bậc nhỏ hơn r. Mỗi phép chia như vậy cần $O(r^2\log^2 p+r\log^3 p)=O(r^2\log^3 p)$ phép tính bit. Như vậy, ta cần tất cả là $O(r^3\log^3 p)=O(\log^3 q)$ phép tính bit, điều phải chứng minh.

Còn phải xét phép tính nâng lên luỹ thừa bậc k. Ta có thể dùng phương pháp bình phương liên tiếp, và như vậy, số phép nhân và bình phương cần thực hiện là $O(\log k)$. Số phép tính bit cần thiết trong trường hợp này là $O(\log k\log^3 q)$. Định lí được chứng minh.

§4. Sự tương tự giữa số nguyên và đa thức.

Sự phát triển của số học, đặc biệt là trong những thập kỉ gần đây, chịu ảnh hưởng rất lớn của sự tương tự giữa số nguyên và đa thức. Nói cách khác, khi có giả thuyết nào đó chưa chứng minh được đối với các số nguyên, người ta cố gắng chứng minh sự kiện tương tự cho các đa thức. Điều đó thường dễ làm hơn, có lẽ nguyên nhân chủ yếu là vì, đối với các đa thức, ta có phép tính đạo hàm, trong khi một khái niệm tương tự chưa có đối với các số nguyên.

Trong tiết này, chúng tôi cố gắng thông qua một vài ví dụ đơn giản, minh họa vai trò quan trọng của sự tương tự nói trên trong các nghiên cứu về số học.

Trước hết, chúng ta thấy rõ, giữa tập hợp các số nguyên và tập hợp các đa thức có những tính chất rất giống nhau sau đây:

1) Các qui tắc cộng, trừ, nhân, chia hoàn toàn như nhau cho cả hai tập hợp.

2) Nếu đối với các số nguyên, ta có các số nguyên tố, thì với các đa thức, ta có các đa thức bất khả quy.

3) Đối với hai số nguyên, cũng như đối với hai đa thức, có thể định nghĩa ước chung lớn nhất. Hơn nữa, trong cả hai trường hợp, ước chung lớn nhất này tìm được bằng thuật toán Euclid.

4) Mỗi số nguyên có phân tích thành các thừa số nguyên tố, mỗi đa thức có phân tích thành các đa thức bất khả quy.

5) Các số hữu tỷ tương ứng với các hàm hữu tỷ.

Chúng tôi dành cho độc giả việc kéo dài bảng danh sách này. Ở đây, chúng tôi sẽ đi vào một vài tương tự khó nhìn thấy hơn.

Ta để ý đến sự tương tự giữa phân tích ra thừa số nguyên tố và phân tích bất khả quy. Nếu giả thiết rằng trường k là đóng đại số, thì mỗi đa thức $Q(x) \in k[x]$ có thể phân tích được dưới dạng sau:

$$Q(x)= P_1^{a_1} P_2^{a_2} \dots P_n^{a_n},$$

trong đó $P_i(x)=(x-\alpha i)$, $\alpha i \in k$.

Như vậy, có thể thấy rằng, trong sự tương tự giữa phân tích bất khả quy và phân tích ra thừa số nguyên tố *các nghiệm của đa thức tương ứng với các ước nguyên tố của số nguyên*. Do đó, số các nghiệm phân biệt của một đa thức có vai trò tương tự như số các ước nguyên tố của một số nguyên. Từ nhận xét đó, ta đi đến định nghĩa sau đây.

Định nghĩa 5.5. Cho a là một số nguyên. Ta định nghĩa *căn của a*, kí hiệu qua $N_0(a)$, là tích các ước nguyên tố của a:

$$N_0(a)=\prod_{p|a} p.$$

Ta sẽ thấy rằng, sự tương tự trên đây cùng với các tính chất của đa thức gợi mở một con đường nhiều hy vọng để đi đến chứng minh định lí Fermat.

Năm 1983, R. C. Mason chứng minh định lí rất đẹp sau đây về các đa thức.

Định lí 5.6. *Giả sử a(t), b(t), c(t) là các đa thức với hệ số phức, nguyên tố cùng nhau từng cặp và thoả mãn hệ thức*

$$a(t)+ b(t)= c(t)$$

Khi đó, nếu kí hiệu qua $n_0(f)$ số nghiệm phân biệt của một đa thức f, thì ta có

$$max\{ deg\ a, deg\ b, deg\ c\} \le n_0(abc)-1.$$

Trước khi đi vào chứng minh định lí, ta chứng minh hệ quả sau đây của định lí Mason.

Hệ quả 5.7. *Không tồn tại các đa thức a, b, c, khác hằng số, nguyên tố cùng nhau, thoả mãn phương trình*

$$a^n + b^n = c^n$$

với $n \ge 3$.

Chứng minh. Gỉa sử các đa thức a, b, c thoả mãn phương trình nói trên. Rõ ràng số nghiệm phân biệt của đa thức $a^n b^n c^n$ không vượt quá $deg\ a + deg\ b + deg\ c$. Áp dụng định lí Mason ta có

$$n\ deg\ a \le deg\ a + deg\ b + deg\ c - 1$$

Viết đẳng thức trên với b, c, rồi cộng từng vế ba bất đẳng thức ta được :

$$n(deg\ a + deg\ b + deg\ c) \le 3\ (deg\ a + deg\ b + deg\ c) - 3$$

Ta có mâu thuẫn nếu $n \geq 3$.

Như vậy, định lí Mason cho ta một chứng minh đơn giản của định lí Fermat cho các đa thức. Sau đây, ta chứng minh định lí Mason.

Chứng minh định lí Mason. Đặt $f = a/b$, $g = a/c$, ta có: $f+g=1$. Lấy đạo hàm hai vế của phương trình này, ta được: $f'+g'=0$. Nhằm mục đích xét số các nghiệm của đa thức, ta xét các thương của đạo hàm và hàm số. Ta có:

$$(f'/f)f+(g'/g)g=0,\ \frac{b}{a}=-\frac{f'/f}{g'/g}.$$

Mặt khác, giả sử $R(t)$ là một hàm hữu tỷ có phân tích sau đây

$$R(t)=\prod(t-\vartheta_i)^{q_i},\quad q_i \in Z.$$

Tính toán đơn giản cho ta:

$$R'/R=\sum\frac{q_i}{t-\vartheta_i}.$$

Bây giờ giả sử a, b, c tương ứng có các nghiệm phân biệt là $\alpha_i, \beta_j, \gamma_k$. Ta có:

$$a(t)=\prod(t-\alpha_i)^{m_i},\ b(t)=\prod(t-\beta_j)^{n_j},\ c(t)=\prod(t-\gamma_k)^{r_k}.$$

Như vậy,

$$\frac{b}{a}=-\frac{f'/f}{g'/g}=-\frac{\sum\frac{m_i}{t-\alpha_i}-\sum\frac{r_k}{t-\gamma_k}}{\sum\frac{n_j}{t-\beta_j}-\sum\frac{r_k}{t-\gamma_k}}$$

Mẫu số chung của các phân số trong phần tử số và mẫu số của thương sau cùng là

$$N_0=\prod(t-\alpha_i)\prod(t-\beta_j)\prod(t-\gamma_k).$$

Đó là một đa thức có bậc là $n_0(abc)$. Như vậy, N_0f'/f và N_0g'/g là các đa thức có bậc không quá $n_0(abc)-1$. Mặt khác ta có:

$$\frac{b}{a}=-\frac{N_0f'/f}{N_0g'/g}$$

Vì a, b nguyên tố cùng nhau nên từ đẳng thức này suy ra bậc của a và bậc của b đều không vượt quá $n_0(abc)-1$. Điều tương tự cũng đúng đối với c do vai trò đối xứng của a, b, c trong phương trình xuất phát. Định lí được chứng minh.

Từ định lí Mason, ta có thể suy ra nhiều hệ thức giữa các đa thức. Chẳng hạn, một trong những hệ quả là định lí sau đây:

Định lí Davenport. *Giả sử $f(t), g(t)$ là các đa thức, sao cho $f^3 \neq g^2$. Khi đó ta có $\deg(f^3-g^2) \geq 1/2 \deg f+1$*

Chúng tôi dành chứng minh định lí này cho độc giả. Khẳng định tương tự đối với các số nguyên vẫn còn chưa được chứng minh. Ta có:

Giả thuyết Hall. Giả sử *x, y* là các số nguyên dương sao cho $x^3 \neq y^2$. Khi đó, với mọi $\varepsilon>0$, tồn tại $C>0$ chỉ phụ thuộc ε sao cho

$$| y^2-x^3|>Cx^{1/2-\varepsilon}$$

Có thể nói thêm rằng, bất đẳng thức trong định lí Davenport là tốt nhất có thể: đối với các đa thức $f(t)=t^6+4t^4+10t^2+6$, $g(t)=t^9+6t^7+21t^5+35t^3+63/2t$ ta có: $deg(f^3-g^2)=1/2degf+1$. Năm 1982, L. V. Danilov cũng đã chứng minh rằng, số mũ 1/2 trong giả thuyết Hall là tốt nhất có thể.

Định lí Mason và tương tự giữa các số nguyên và đa thức đã gợi ý cho giả thuyết sau đây:

Giả thuyết abc (Oesterlé, 1986). Giả sử *a, b, c* là các số nguyên, nguyên tố cùng nhau và thoả mãn hệ thức $a+b=c$. Khi đó, với mọi $\varepsilon>0$, tồn tại số *C* sao cho

$$\max(|a|, |b|, |c|)<CN^{1+\varepsilon},$$

trong đó $N=\prod_{p|abc} p$ là căn của *abc*.

Hoàn toàn tương tự như trên, từ giả thuyết "*abc*" có thể suy ra *Định lí Fermat tiệm cận*: với *n* đủ lớn, phương trình Fermat không có nghiệm nguyên.

Nhận xét. Giả sử *p* là một ước nguyên tố nào đó của một trong các số *a, b, c,* chẳng hạn $p|a$. Khi đó, nếu *p* lớn thì trong phân tích của *a* ra thừa số nguyên tố, *p* phải có số mũ tương đối nhỏ (để $|a|$ không vượt quá xa căn của *abc* theo giả thuyết). Điều này cũng giải thích tại sao phương trình Fermat không có nghiệm với bậc đủ lớn: khi đó, mọi ước nguyên tố của a^n, b^n, c^n sẽ tham gia với bậc quá lớn.

Trên đây là một ví dụ về sự tương tự giữa các giả thuyết đối với các số và các đa thức. Khi nghiên cứu một vấn đề nào đó đặt ra đối với các số, người ta thường nghiên cứu đồng thời tương tự của nó trên trường hàm. Phần lớn các giả thuyết của số học được chứng minh trước hết trên trường hàm. Như ta đã thấy trong chứng minh định lí Mason, điều quan trọng ở đây là có phép tính đạo hàm.

Gần đây, Manin (1992) đặt ra vấn đề: nếu như các số nguyên tương ứng với các đa thức một biến, thì các đa thức nhiều biến tương ứng với đối tượng nào? Câu hỏi đó dẫn đến việc xây dựng những "tích" mới của các "lược đồ" Spec Z. Đây là một hướng nghiên cứu đang phát triển mạnh, và nội dung của nó vượt ngoài khuôn khổ của cuốn sách này.

Bài tập và tính toán thực hành chương 5

I. Bài tập

5.1. Giả sử K là trường đặc trưng p. Chứng minh rằng, đối với các phần tử của trường K, ta có:

$$(a+b)^p=a^p+b^p.$$

5.2. Xây dựng trường F_9 từ trường F_3 bởi các đa thức sau:

1) x^2+1

2)x^2-x-1.

5.3. Dùng thuật toán EP để tìm ƯCLN của các đa thức P,Q trong trường F_p, và biểu diễn dạng $d=uP+vQ$:

1) $P=x^3+x+1$, $Q=x^2+x+1$, $p=2$.

2) $P=x^6+x^5+x^4+x^3+x^2+x+1$, $Q=x^4+x^2+x+1$, $p=2$.

3) $P=x^3-x+1$, $Q=x^2+1$, $p=3$.

4) $x^5+x^4+x^3-x^2-x+1$, $Q=x^3+x^2+x+1$, $p=3$.

5) $x^5+88x^4+73x^3+83x^2+51x+67$, $Q=x^3+97x^2+40x+38$, $p=101$.

5.4. Với mỗi $d \leq 6$, tìm tất cả các đa thức bất khả quy bậc d trên trường F_2.

5.5. Những đa thức nào trong $F_p[x]$ có đạo hàm đồng nhất bằng 0?

5.6. Chứng minh rằng từ giả thuyết "*abc*" suy ra định lí Fermat tiệm cận.

5.7. Chứng minh định lí Davenport.

5.8. Cho f, g là các đa thức với hệ số nguyên, sao cho f^3-g^4 không đồng nhất bằng 0. Chứng minh rằng

$$\deg(f^3-g^4) \geq 5/3\deg g+1.$$

Phát biểu kết luận tương tự cho các số nguyên.

5.9. Dựa vào định lí Mason, tìm những hệ thức mới liên quan đến bậc của các đa thức hệ số nguyên (tương tự bài tập trên đây). Thử phát biểu và chứng minh kết luận tương tự dối với các số nguyên.

5.10. Thử đưa ra một định nghĩa về đạo hàm của một số nguyên.

II. Thực hành trên máy tính

II. 1. Thực hành tính số đa thức bất khả quy bậc *d* trên trường hữu hạn

Để tính số đa thức bất khả quy bậc n trên trường hữu hạn có đặc số *p* ta thực hiện dòng lệnh như sau;

```
[>mipolys(n, p);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện số đa thức cần tìm.

Thí dụ: Tính số các đa thức bất khả quy bậc 6 trên trường có đặc số 2.

Ta thực hiện lệnh sau:

```
[>mipolys(6, 2);
```

```
                              9
```

Như vậy có 9 đa thức bất khả quy trên trường F_2.

II. 2. Thực hành tìm ước chung lớn nhất của các đa thức trên trường hữu hạn

Cho *P, Q* là các đa thức biến *x* với hệ số trên trường hữu hạn có đặc số *p*. Để tìm ước chung lớn nhất *D* của *P* và *Q* và biểu diễn dưới dạng D=sP+tQ ta thực hiện dòng lệnh sau:

```
[> Gcdex(P,Q,x,'s','t') mod p;
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện một đa thức, đó chính là ước chung lớn nhất của *P, Q*. Tiếp tục thực hiện lệnh:

```
[>s,t;
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện hai đa thức *s, t* cần tìm.

Chú ý lệnh "Gcdex" chữ "G" là chữ viết hoa.

Thí dụ1: Tìm ước chung lớn nhất *D* của các đa thức *P, Q* trên trường F_2 và biểu diễn dưới dạng D=sP+tQ, trong đó $P=x^3+x+1$, $Q=x^2+x+1$.

Ta thực hiện dòng lệnh:

```
[> Gcdex(x^3+x+1,x^2+x+1,x,'s','t') mod 2;
```

```
                              1
```

```
[>s,t;
```

```
                          1+x, x^2
```

Vậy $1=(1+x)P+x^2Q$.

Thí dụ2: Tìm ước chung lớn nhất *D* của các đa thức *P, Q* trên trường F_{101} và biểu diễn dưới dạng D = sP + tQ, trong đó $x^5 + 88x^4 + 73x^3 + 83x^2 + 51x + 67$, $Q=x^3+97x^2+40x+38$

Ta thực hiện dòng lệnh:

```
[>Gcdex(x^5+88*x^4+73*x^3+83*x^2+51*x+67,x^3+97*x^2+40*x
+38,x,'s','t') mod 101;
```

```
                                    x + 78
[> s,t;
                          50x + 20, 51x^3 + 26x^2 + 27x + 4
```

Vậy $x+78=(50x+20)P+(51x^3+26x^2+27x+4)Q$.

II. 3. Thực hành tìm ước chung lớn nhất, bội chung nhỏ nhất của các đa thức trên trường hữu tỷ

Cho *P, Q* là các đa thức biến *x* với hệ số trên trường hữu tỷ.

1. Để tìm ước chung lớn nhất *D* của *P* và *Q* ta thực hiện dòng lệnh sau:

```
[> gcd(P,Q);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả, đó chính là ước chung lớn nhất của *P, Q*.

Thí dụ1: Tìm ước chung lớn nhất *D* của các đa thức *P, Q* trên trường hữu tỷ trong đó $P=x^2-y^2$, $Q=x^3-y^3$.

Ta thực hiện dòng lệnh:

```
[> gcd(x^2-y^2,x^3-y^3);
                                    -y+x
```

Vậy ước chug lớn nhất của *P* và *Q* là -y+x

2. Để tìm bội chung nhỏ nhất của *P, Q* ta thực hiện dòng lệnh như sau:

```
[> lcm(P,Q);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ xuất hiện kết quả, đó chính là bội chung nhỏ nhất của *P,Q*.

Thí dụ 2: Tìm bội chung nhỏ nhất của các đa thức *P, Q* trên trường hữu tỷ trong đó $P=x^2-y^2$, $Q=x^3-y^3$.

Ta thực hiện dòng lệnh:

```
[> lcm(x^2-y^2,x^3-y^3);
                                -yx^3+y^4-x^4+xy^3
```

Vậy bội chung nhỏ nhất của *P* và *Q* là $-yx^3+y^4-x^4+xy^3$

Chương 6

VÀI ỨNG DỤNG VÀO LÍ THUYẾT MẬT MÃ

Cho đến khoảng cuối những năm 70, số học vẫn được xem là một trong những ngành lí thuyết thuần tuý nhất của toán học, vì hầu như không có ứng dụng thực tế. Quan niệm đó thay đổi hẳn sau khi số học được áp dụng để xây dựng những hệ mật mã khoá công khai. Các lí thuyết mới của số học, đặc biệt là số học thuật toán, tìm thấy những ứng dụng trực tiếp vào thực tiễn. Vì thế chúng tôi dành một chương trình bày những điểm cơ bản của lí thuyết mật mã, để qua đó, độc giả có thể thấy được vai trò quan trọng của những vấn đề xét đến trong lí thuyết số thuật toán, như vấn đề độ phức tạp của các thuật toán phân tích một số nguyên dương ra thừa số, hay vấn đề kiểm tra nguyên tố.

§1. Mã Ceasar.

Có thể nói, mật mã đã có từ thời cổ đại. Người ta cho rằng, người đầu tiên áp dụng mật mã một cách có hệ thống để đảm bảo bí mật thông tin quân sự là nhà quân sự thiên tài của La Mã cổ đại, Julius Ceasar. Sự phát triển của xã hội dẫn đến việc ngày nay mật mã không những chỉ được dùng trong bí mật quân sự và ngoại giao, mà còn dùng, và có thể chủ yếu là dùng trong bí mật kinh tế, thương mại. Vì thế xuất hiện những đòi hỏi mới đối với các hệ mật mã hiện đại, khác về nguyên tắc so với mật mã thường dùng trước đây. Khác với hoạt động quân sự hoặc ngoại giao, trong hoạt động kinh doanh, số lượng đơn vị phải cùng trao đổi thông tin thường là rất lớn. Thậm chí, những người chưa hề quen biết nhau cũng có nhu cầu trao đổi những thông tin mật với nhau. Bởi thế, những hệ thống mật mã xây dựng theo nguyên tắc cũ khó có thể thích hợp: trong các hệ mã đó, khi đã biết khoá lập mã, ta dễ dàng tìm ra khoá giải mã. Hiển nhiên, muốn gửi một thông báo mật cho một đối tượng nào đó, ta cần phải biết khoá lập mã của họ, vì thế, những người cùng dùng một hệ mã đều biết hết bí mật của nhau. Khi một bí mật có quá nhiều người biết thì không còn là bí mật nữa. Các hệ thống mật mã hiện đại, *mật mã khoá công khai*, khắc phục được những nhược điểm đó: mỗi người tham gia trong hệ thống chỉ cần giữ bí mật khoá giải mã của mình, trong khi khoá lập mã được thông báo công khai. Việc biết khoá lập mã không cho phép tìm ra khoá giả mã trong một thời gian chấp nhận được, ngay cả khi sử dụng những máy tính hiện đại nhất. Những mật mã khoá công khai tìm thấy đầu tiên là những mật mã dùng hàm số học.

Có một điều hết sức thú vị là, nói cho cùng, những hệ mật mã hiện đại cũng chỉ là sự cải tiến mật mã của Ceasar! Vì thế chúng tôi bắt đầu việc trình bày mật mã Ceasar.

Trước hết chúng ta cần thống nhất một số danh từ.

Văn bản tức là thông báo cần chuyển, được viết bằng ngôn ngữ thông thường. Ở đây, ta sẽ xem các văn bản đều được viết bằng tiếng Việt.

Việc chuyển thông báo đó thành dạng mật mã được gọi là *mã hóa*.

Bản đã mã hoá của văn bản được gọi là *văn bản mật*.

Giải mã tức là chuyển một văn bản mật thành văn bản ban đầu.

Ceasar chuyển thông báo mật bằng cách sau đây. Trước tiên, lập tương ứng mỗi chữ cái với một số. Nhờ bảng tương ứng đó, ta có thể chuyển một văn bản thành dạng chữ số. Sau đó ta cộng thêm 3 vào mỗi chữ số nhận được. Lại nhờ bảng tương ứng giữa chữ và số, ta biến bảng chữ số mới này về dạng chữ viết. Như vậy ta nhận được một văn bản mật cần chuyển đi. Đây là quá trình mã hoá.

Khi nhận được văn bản mật, ta giải mã bằng cách biến nó thành dạng chữ số nhờ bảng tương ứng giữa chữ và số, sau đó trừ đi 3 ở mỗi chữ số và lại chuyển nó về dạng chữ để lại có văn bản ban đầu.

Chú ý rằng khi phép cộng hoặc trừ đi 3 đưa ta vượt khỏi giới hạn của bảng tương ứng, ta thay số đó bằng thặng dư dương bé nhất modulo số các phần tử của bảng tương ứng giữa chữ và số.

Sau đây ta sẽ xét trên một ví dụ cụ thể.

Trước hết ta lập tương ứng các chữ cái với các số theo bảng sau:

a	ă	â	b	c	d	đ	e	ê	g	h
1	2	3	4	5	6	7	8	9	10	11

i	k	l	m	n	o	ô	ơ	p	q
12	13	14	15	16	17	18	19	20	21

r	s	t	u	ư	v	x	y
22	23	24	25	26	27	28	29

Bảng 1

Dĩ nhiên ta có thể thêm các số để chỉ dấu, nhưng để đơn giản , ở đây ta tạm thời viết các văn bản không dấu.

Như vậy mã Ceasar được thành lập theo công thức sau:

$$C \equiv P+3 (\text{mod } 29) \qquad (6.1)$$

trong đó P là chữ số trong văn bản, còn C là chữ số tương ứng trong văn bản mật. Chẳng hạn ta muốn mã hoá thông báo sau đây:

LY THUYÊT MÂT MA KHÔNG CO GI KHO

Trước hết nhằm nâng cao tính bảo mật, ta tách thông báo thành từng nhóm 5 chữ cái, để tránh việc một số từ của thông báo dễ bị phát hiện căn cứ vào số chữ cái. Như vậy thông báo cần mã hoá là:

LYTHU YÊTMÂ TMAKH ÔNGCO GIKHO

Nhờ bảng 1, ta chuyển thông báo thành dạng chữ số:

14 29 24 11 25 29 8 24 15 1 24 15 1 13 11 18 16 10 5 18 10 12 13 11 18

Ap dung công thức (6.1), bảng chữ số trên được chuyển thành:

17 3 27 14 28 3 11 27 18 3 27 18 4 16 14 21 19 13 8 21 13 15 16 14 21

Để có văn bản mật, ta chỉ cần chuyển lại thành dạng chữ cái theo Bảng 1:

OÂVLU ÂHVÔÂ VÔBNL QƠKEQ KMNLQ

Số 3 trong công thức (6.1) được gọi là khoá của mã Ceasar, vì nó được dùng để mã hoá cũng như giải mã.

Ta cũng có thể lập một hệ mật mã mới bằng cách thay số 3 trong công thức (6.1) bằng một số *k* tuỳ ý khác giữa 1 và 29:

$$C \equiv P+k \pmod{29} \qquad (6.2)$$

Trong trường hợp này, khoá của mã là *k*. Việc mã hoá và giải mã được tiến hành hoàn toàn tương tự như trên.

Ta có thể lập mã tổng quát hơn chút ít bằng cách thay công thức (6.2) bởi công thức sau đây:

$$C \equiv aP+b \pmod{29},$$

trong đó *a, b* là các số nguyên, và (a, 29)=1. Những mã như vậy được gọi là *mã biến đổi aphin*. Việc giải mã được tiến hành bằng cách giải phương trình đồng dư (6.2), khi đã biết *c, a, b*.

Phân tích sau đây cho thấy tính bảo mật của mã Ceasar là không cao. Khi bắt được một văn bản mật, người ta có thể dựa vào tần suất xuất hiện của các chữ cái để đoán ra khoá của mã. Chẳng hạn nếu chữ *a* nói chung xuất hiện nhiều nhất trong các văn bản thì chữ cái nào có mặt nhiều nhất trong văn bản mật có nhiều khả năng là chữ *a,* từ đó đoán ra khoá. Hơn nữa, chỉ có 29 cách khác nhau để chọn khoá cho loại mã nói trên, nên để dàng tìm ra khoá của mã, nhất là khi áp dụng máy tính. Đối với mã biến đổi aphin, chỉ cần dựa vào tần suất xuất hiện từ để tìm ra hai chữ cái tương ứng với 2 chữ nào đó trong văn bản mật, ta có thể tìm ra *a, b* bằng cách giải hệ hai phương trình đồng dư. Ngoài ra, việc giải những hệ mã biến đổi aphin cũng quá dễ dàng đối với máy tính.

Như vậy, với những yêu cầu về bảo mật cao hơn, người ta phải dùng những hệ mật mã phức tạp hơn. Sau đây là một vài hệ mã thường dùng, từ đơn giản đến phức tạp.

§2. Mã khối.

Mã khối xuất hiện nhằm chống lại việc sử dụng tần suất xuất hiện của các chữ cái trong văn bản để dò ra khoá giải mã. Khác với các hệ mã trình bày ở mục trên, ta không mã hoá từng chữ cái của văn bản, mà mã hoá từng khối chữ cái. Trước tiên ta xét trường hợp mã khối 2 chữ. Để dễ hiểu ta xét ví dụ sau đây.

Giả sử thông báo cần mã hoá là

KHÔNG CO ĐIÊU BI MÂT NAO GIU ĐƯƠC LÂU

Trước hết ta tách thông báo trên thành khối hai chữ:

KH ÔN GC OĐ ÊU BI MÂ TN AO GI UĐ ƯƠ CL ÂU

Sau đó các chữ cái được chuyển thành các chữ số tương ứng:

13 11 18 15 10 5 17 7 9 25 4 12 16 3 24 15 1 17 1012 26 19 5 14 3 25

Với mỗi mã khối hai chữ, ta chọn một ma trận cấp hai làm khoá của mã. Chẳng hạn ma trận

$$A=\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$$

Khi đó các khối hai chữ số P_1P_2 trong văn bản được chuyển thành các khối hai chữ số C_1C_2 trong văn bản mật theo công thức sau đây:

$$C_1 \equiv 23P_1+11P_2 (\text{mod } 29)$$
$$C_2 \equiv 9P_1+12P_2 (\text{mod } 29) \qquad (6.3)$$

Như vậy thông báo trên đây đã được chuyển thành:

14 17 28 23 24 5 4 5 18 4 21 6 21 19 7 10 14 2 24 27 8 10 25 8

Trở lại các chữ cái tương ứng, ta được văn bản mật:

LO XS TC BC ÔB QD TD QƠ ĐG LI TV EG UE

Để giải mã, ta cần giải hệ phương trình đồng dư (6.3) để tìm P_1,P_2. Điều đó thực hiện được nhờ định lí sau đây:

Định lí 6.1. *Cho hệ phương trình đồng dư*

$$ax+by \equiv r(\text{mod } m)$$
$$cx+dy \equiv s(\text{mod } m)$$

Đặt $\Delta = ad-bc \ (\text{mod } m)$. *Khi đó, nếu* $(\Delta, m)=1$ *thì hệ phương trình đang xét tồn tại nghiệm duy nhất modulo m, cho bởi công thức sau:*

$$x \equiv \Delta^{-1}(dr-bs) \ (\text{mod } m),$$
$$y \equiv \Delta^{-1}(as-cr) \ (\text{mod } m),$$

trong đó Δ^{-1} *là nghịch đảo của* Δ *modulo m.*

Định lí trên đây được chứng minh hoàn toàn tương tự như trong đại số tuyến tính (chỉ cần thay điều kiện $(\Delta, m)=1$ bởi điều kiện $\Delta \neq 0$).

Trong ví dụ trên đây, $\Delta \equiv 3 (mod\ 29)$, $\Delta^{-1} \equiv 10 (mod\ 29)$. Như vậy, khi có khối C_1C_2 trong văn bản mật và đã biết mã khoá là ma trận A, ta tìm được khối chữ tương ứng trong văn bản là P_1P_2 theo công thức sau:

$$P_1=10(12C_1-11C_2) \equiv 4C_1+6C_2 (\text{mod } 29)$$

$$P_2=10(23C_2-9C_1) \equiv 26C_1+27C_2 (\text{mod } 29).$$

Tóm lại, việc mã hoá và giải mã được tiến hành nhờ các công thức:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}; \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 26 & 27 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

Như vậy, việc sử dụng mã khối đã nâng cao rất nhiều tính bảo mật. Tuy vậy, khoá của mã vẫn có thể bị khám phá nhờ việc nghiên cứu tần suất xuất hiện của các khối chữ cái. Chẳng hạn nếu ta dùng mã khối hai chữ, thì có cả thảy $29^2=641$ khối trong tiếng Việt, và như vậy vẫn còn khả năng khám phá ra khoá của mã nhờ các máy tính hiện đại. Trong trường hợp ta sử dụng mã khối với những khối nhiều chữ cái, việc tìm ra khoá bằng tần suất các khối chữ trên thực tế là không sử dụng được: Chẳng hạn khi dùng mã khối 10 chữ cái, số khối chữ sẽ là 29^{10}, vượt quá khả năng thăm dò tần suất xuất hiện các khối chữ đó trong ngôn ngữ.

Các mã khối n chữ cái được lập và giải hoàn toàn tương tự như trên, trong đó các ma trận C,P là các ma trận n cột, A là ma trận vuông cấp n. Ma trận nghịch đảo của A tồn tại khi định thức của A nguyên tố cùng nhau với 29, và ma trận P sẽ được tính bằng quy tắc Kramer như trong đại số tuyến tính (chỉ cần thay dấu = bởi $\equiv (mod\ 29)$).

Sau đây ta trình bày một loại hệ mã mới, một mặt nó có tính bảo mật rất cao, Mặt khác là cơ sở cho những hệ mã hoàn toàn mới: các hệ mã khoá công khai.

§3. Mã mũ.

Hệ mã này được Pohlig và Hellman đưa ra năm 1978.

Giả sử p là một số nguyên tố lẻ, và giả sử khoá lập mã e là một số nguyên dương sao cho $(e,p-1)=1$. Cũng như trước đây, để mã hoá một thông báo, trước tiên ta chuyển các chữ cái thành dạng các chữ số tương ứng (thêm số 0 vào trước những số có một chữ số). Ta dùng bảng sau đây:

a	ă	â	b	c	d	đ	e	ê	g	h
01	02	03	04	05	06	07	08	09	10	11

i	k	l	m	n	o	ô	ơ	p	q
12	13	14	15	16	17	18	19	20	21

r	s	t	u	ư	v	x	y
22	23	24	25	26	27	28	29

Sau đó ta nhóm các số nhận được thành từng nhóm $2m$ chữ số theo nguyên tắc sau: $2m$ là số nguyên chẵn lớn nhất sao cho mọi số tương ứng với m chữ cái (xét như là một số nguyên có $2m$ chữ số) đều nhỏ hơn p. Để dễ hiểu, ta giả sử p là số nguyên tố trong khoảng $2929<p<292929$. Mỗi chữ cái được biểu diễn bằng một số không quá 29. Một số có m chữ cái sẽ được biểu diễn bằng một số không vượt quá m lần số 29 viết liên tiếp. Như vậy, để đảm bảo số đó luôn luôn nhỏ hơn p, m chỉ có thể là 1 hoặc 2. Ta lấy $m=2$.

Đối với một khối P trong văn bản (là một số $2m$ chữ số), ta lập khối C tương ứng trong văn bản mật theo công thức sau:

$$C \equiv P^e \pmod{p},\ 0 \le P < p$$

Văn bản mật sẽ chứa những khối chữ số là các số nguyên nhỏ hơn p.

Ví dụ. Giả sử số nguyên tố sử dụng để tiến hành lập mã là $p=2939$ và khoá lập mã $e=31$, như vậy $(e,p-1)=(31,2938)=1$.

Ta cần mã hoá thông báo sau:

ĐI HA NÔI NGAY

Trong trường hợp này, $m=2$, và ta nhóm văn bản nhận được khi chuyển sang chữ số thành nhóm bốn chữ số:

0712 1101 1618 1216 1001 2928

Chú ý rằng, để khối cuối cùng đủ bốn chữ số, ta thêm chữ X trong văn bản, điều này không gây nhầm lẫn khi đọc thông báo (dĩ nhiên có thể thay chữ X bằng bất cứ chữ cái naò không gây hiểu nhầm).

Tiếp theo, ta chuyển các khối P trong văn bản thành các khối C trong văn bản mật theo công thức sau:

$$C \equiv P^{31} \pmod{2939},\ 0<C<2633$$

Chẳng hạn, để mã hoá khối đầu tiên, ta tính:

$$C \equiv 0712^{31} \pmod{2939}$$

Để tính được C một cách nhanh chóng, ta dùng thuật toán bình phương liên tiếp đã xét trong chương 5.

Trước tiên, ta viết 31 dưới dạng cơ số 2: $31=(11111)_2$. Tính toán đơn giản cho ta:

$$721^2 \equiv 1436,\ 712^4 \equiv 1857,\ 712^8 \equiv 1002,\ 712^{16} \equiv 1805 \pmod{2939}.$$

Từ biểu diễn của 31 dưới dạng cơ số 2, ta được:

$$712^{31} \equiv (712.1436.1857.1805) \equiv 898 \pmod{2939}.$$

Sau khi mã hoá toàn bộ văn bản, ta nhận được văn bản mật cần chuyển là:

898 2674 1003 746 1786 2614

Để giải mã một khối C trong văn bản mật, ta cần biết khoá giải mã d. Đó là số d thoả mãn $de \equiv 1 (mod\ p\text{-}1)$, có nghĩa là d là một nghịch đảo của e modulo p-1. Nghịch đảo đó tồn tại do giả thiết $(e,p\text{-}1)=1$. Để tìm lại được khối C trong văn bản, ta chỉ việc nâng khối C lên luỹ thừa d *modulo* p. Thật vậy,

$$C^d \equiv (P^e)^d \equiv P^{de} \equiv P^{k(p-1)+1} \equiv P(\text{mod } p)$$

trong đó $de=k(p\text{-}1)+1$ đối với số nguyên k nào đó, bởi vì $de \equiv 1 (mod\ p\text{-}1)$.

Ví dụ. Để giải mã một khối trong văn bản mật được mã hoá bằng cách sử dụng modulo p=2938 và khoá lập mã e=31, ta cần tìm số nghịch đảo của e=31 modulo p-1=2938. Thuật toán Euclid mở rộng giúp ta dễ dàng tìm được d. Thật vậy, theo các kí hiệu của thuật toán Euclid mở rộng, ta đặt: u=2938, v=31. Tính toán theo thuật toán đó, ta được kết quả sau đây:

q	u_1	u_2	u_3	v_1	v_2	v_3
-	1	0	2938	0	1	31
94	0	1	31	1	-94	24
1	1	-94	24	-1	95	7
3	-1	95	7	4	-379	3
2	4	-379	3	-9	853	1
3	-9	853	1	31	-2938	0

Như vậy, ta có: 31.853-9.2938=1, và d=853

Để giải mã khối C ta dùng công thức

$$P \equiv C^{853} (\text{mod } 2938).$$

Độ phức tạp của thuật toán lập mã và giải mã đối với mã mũ.

Với một khối P trong văn bản, ta mã hoá bằng cách tính $P^e (mod\ p)$, số các phép tính bit cần thiết là $O((log_2\ p)^3)$. Để giải mã trước hết ta phải tìm nghịch đảo d của e modulo p-1. Điều này thực hiện được với $O(log^3\ p)$ phép tính bit, và chỉ cần làm một lần. Tiếp theo đó, để tìm lại được khối P của văn bản từ khối C của văn bản mật, ta chỉ cần tính thặng dư nguyên dương bé nhất của C^d *modulo* p: số các phép tính bit đòi hỏi là $O((log_2 p)^3)$.

Như vậy, thuật toán lập mã và giải mã được thực hiện tương đối nhanh bằng máy tính.

Tuy nhiên ta sẽ chứng tỏ rằng, việc giải mã một văn bản mật được mã hoá bằng mũ nói chung không thể làm được nếu như không biết khoá *e*. Thật vậy, giả sử ta đã biết số nguyên tố *p* dùng làm modun khi lập mã, và hơn nữa, giả sử đã biết khối *C* nào đó trong văn bản mật tương ứng với khối *P* trong văn bản, tức là ta đã biết một đồng dư thức

$$C \equiv P^e \pmod{p}$$

Vấn đề còn lại là xác định *e* từ công thức trên. Số *e* thoả mãn điều kiện đó được gọi là l*ôgarít cơ số P của C modulo p*. Có nhiều thuật toán khác nhau để tìm lôgarit cơ số đã cho modulo một số nguyên tố. Thuật toán nhanh nhất được biết hiện nay đòi hỏi khoảng $exp(\sqrt{\log p \log\log p})$ phép tính bit. Để tìm logarit modulo một số nguyên tố có *n* chữ số thập phân, các thuật toán nhanh nhất cũng đòi hỏi số phép tính bit xấp xỉ số phép tính bit cần dùng khi phân tích một số nguyên *n* chữ số thành thừa số. Như vậy, nếu làm việc với các máy tính có tốc độ 1 triệu phép tính trong một giây, khi *p* có khoảng 100 chữ số thập phân, việc tìm logarit modulo *p* cần khoảng 74 năm, còn trong trường hợp *p* có khoảng 200 chữ số, thời gian cần thiết là 3,8 tỷ năm!

Cần phải lưu ý rằng, có những trường hợp việc tìm ra logarit modulo *p* được thực hiện bằng những thuật toán nhanh hơn rất nhiều. Chẳng hạn khi p-1 chỉ có những ước nguyên tố nhỏ, tồn tại những thuật toán đặc biệt cho phép tính logarit modulo *p* với $O(log^2 p)$ phép tính bit. Rõ ràng những số nguyên tố như vậy không thể dùng để lập mã. Trong trường hợp đó, ta có thể lấy *q* với $p=2q+1$, nếu số *q* cũng là số nguyên tố (khi đó *q*-1 không thể có các ước nguyên tố nhỏ).

Mã mũ và hệ thống có nhiều cá thể tham gia.

Một trong những ưu điểm của hệ mã mũ là trong một hệ thống có nhiều cá thể cùng tham gia trao đổi thông tin, từng cặp cá thể hoặc từng nhóm nhỏ cá thể vẫn có khả năng sử dụng khoá mật mã đang dùng để tạo những khoá mật mã chung, bí mật đối với các cá thể còn lại của hệ thống.

Giả sử *p* là một số nguyên tố lớn và *a* là một số nguyên, nguyên tố cùng nhau với *p*. Mỗi cá thể trong hệ thống chọn một số *k* nguyên tố cùng nhau với *p*-1 làm khoá cho mình. Khi hai cá thể với các khoá k_1, k_2 muốn lập một khoá chung để trao đổi thông tin, cá thể thứ nhất gửi cho cá thể thứ hai số nguyên y_1 tính theo công thức:

$$y_1 \equiv a^{k_1} \pmod{p},\ 0 < y_1 < p$$

Cá thể thứ hai sẽ tìm ra khoá chung *k* bằng cách tính

$$k \equiv y_1^{k_2} \equiv a^{k_1 k_2} \pmod{p},\ 0 < k < p$$

Tương tự cá thể thứ hai gửi cho cá thể thứ nhất số nguyên y_2

$$y_2 \equiv a^{k_2} \pmod{p},\ 0 < y_2 < p$$

và cá thể thứ nhất tìm ra khoá chung *k* theo công thức

$$k \equiv y_2^{k_2} \equiv a^{k_1 k_2} \pmod{p}$$

Ta lưu ý rằng, trong cách lập khoá chung trên đây, các cá thể thứ nhất và thứ hai không cần biết khoá mật mã của nhau, mà chỉ sử dụng khoá mật riêng của mình. Mặt khác, các cá thể còn lại của một hệ thống cũng không thể tìm ra khoá chung k đó trong một thời gian chấp nhận được, vì để làm việc đó, họ phải tính logarit modulo p.

Trên đây là cách lập khoá chung của hai cá thể. Hoàn toàn tương tự như vậy, từng nhóm các thể có thể lập khoá chung.

§4. Các hệ mật mã khoá công khai.

Trong tất cả các hệ mật mã trình bày trên đây, các khoá lập mã đều phải được giữ bí mật, vì nếu khoá lập mã bị lộ thì người ta có thể tìm ra khoá giải mã trong một thời gian tương đối ngắn. Như vậy nếu trong một hệ thống có nhiều cặp cá thể hoặc nhiều nhóm cá thể cần trao đổi thông tin mật với nhau, số khoá mật mã chung cần giữ bí mật là rất lớn, và như vậy, khó có thể bảo đảm được. Hệ mã mà chúng ta nghiên cứu dưới đây được lập theo một nguyên tắc hoàn toàn mới, trong đó việc biết khoá lập mã không cho phép tìm ra khoá giải mã trong một thời gian chấp nhận được. Vì thế, mỗi cá thể chỉ cần giữ bí mật khoá giải mã của riêng mình, trong khi khoá lập mã được thông báo công khai. Trong trường hợp một trong các cá thể bị lộ khoá giải mã của mình, bí mật của các cá thể còn lại không hề bị ảnh hưởng. Lí do của việc có thể xây dựng những hệ mã như vậy chính là điều ta đã nói đến khi xét các hệ mã mũ: độ phức tạp của thuật toán tìm logarit modulo p là quá lớn.

Trước hết, ta nói sơ qua về nguyên tắc của các hệ mã khoá công khai. Giả sử trong hệ thống đang xét có n cá thể cùng trao đổi các thông tin mật. Mỗi các thể chọn cho mình một khoá lập mã k và một công thức mã hoá $E(k)$, được thông báo công khai. Như vậy có n khoá lập mã công khai $k_1, k_2,...,k_n$. Khi cá thể thứ i muốn gửi thông báo cho cá thể thứ j, cũng như trước đây, mỗi chữ trong thông báo được chuyển thành số, nhóm thành từng khối với độ dài nào đó. Sau đó, mỗi khối P trong văn bản được mã hoá bằng khoá lập mã $E(k_j)$ của cá thể thứ j (đã thông báo công khai), và gửi đi dưới dạng $C=E(k_j)(P)$. Để giải mã thông báo này, cá thể thứ j chỉ cần dùng khoá giải mã (bí mật riêng cho mình) D_{k_j}

$$D_{k_j}(\mathrm{C}) = D_{k_j}\, E_{k_j}(P) = P,$$

bởi vì D_{k_j} và E_{k_j} là các khoá giải mã và lập mã của cùng cá thể thứ j. Các cá thể trong hệ thống, nếu nhận được văn bản mật, cũng không thể nào giải mã, vì việc biết khoá lập mã E_{k_j} không cho phép tìm ra khoá giải mã D_{k_j}.

Để cụ thể hoá nguyên tắc vừa trình bày, ta xét ví dụ trên hệ mã khoá công khai được tìm thấy đầu tiên năm 1978 bởi Rivest, Shamir và Adleman (xem[RSA]) (thường được gọi là hệ mã RSA).

Hệ RSA được xây dựng trên cơ sở mã mũ, trong đó khoá là cặp (e,n), gồm số mũ e và *modun* n. Số n được dùng ở đây là tích của hai số nguyên tố rất lớn nào đó, $n=pq$,

sao cho $(e,\phi(n))=1$, trong đó $\phi(n)$ là hàm Euler. Để mã hoá một thông báo, trước tiên ta chuyển các chữ cái thành các số tương ứng và nhóm thành các khối với độ dài lớn nhất có thể (tuỳ thuộc khả năng tính toán) với một số chẵn chữ số. Để mã hoá một khối P trong văn bản, ta lập khối C trong văn bản mật bằng công thức:

$$E(P) \equiv C \equiv P^e (\text{mod } n),\ 0<C<n.$$

Quá trình giải mã đòi hỏi phải biết được một nghịch đảo d của e modulo $\phi(n)$. Nghịch đảo này tồn tại theo điều kiện $(e,\ \phi(n))=1$.

Muốn giải mã một khối C trong văn bản mật, ta tính

$$D(C) \equiv C^d \equiv (P^e)^d \equiv P^{ed} \equiv P^{k\phi(n)+1} \equiv (P^{\phi(n)})^k P \equiv P(\text{mod } n).$$

trong đó $ed=k\phi(n)+1$ đối với số nguyên k nào đó, vì $ed \equiv 1(\text{mod } \phi(n))$, và do định lí Euler ta có: $P^{\phi(n)} \equiv 1(\text{mod } p)$, khi $(P,n)=1$ (chú ý rằng, xác suất để P và n không nguyên tố cùng nhau là hết sức nhỏ, xem bài tập 6.7). Cặp (d,n) như vậy được gọi là khoá giải mã.

Để minh hoạ, ta xét một ví dụ đơn giản, lấy $n=53.61=3233$ và $e=17$. Trong trường hợp đó ta có $(e,\ \phi(n))=(17,52.63)=1$. Giả sử ta cần mã hoá thông báo sau:

ĐA GƯI TIÊN

Trước tiên ta chuyển các chữ cái trong văn bản thành các số tương ứng và nhóm chúng thành từng khối 4 chữ số. Ta có:

0701 1026 1224 1209 1628

Ta mã hoá các khối nhờ công thức

$$C \equiv P^{17} (\text{mod } 3233)$$

Ta lại dùng phương pháp bình phương liên tiếp. Chẳng hạn, đối với khối đầu tiên, ta nhận được:

$$(701)^{17} \equiv 140(\text{mod } 3233)$$

Mã hoá toàn bộ văn bản, ta được văn bản mật sau đây:

140 721 1814 1819 361

Khi nhận được văn bản mật này, để giải mã, ta phải tìm một nghịch đảo d của e modulo $\phi(3233)$. Ta có $\phi(53.61)=52.60=3120$. Dùng thuật toán Euclid mở rộng, ta tính được $d=2753$. Như vậy, để giải mã khối C ta dùng công thức

$$P \equiv C^{2753} (\text{mod } 3233),\ 0 \le P \le 3233$$

Có thể thử lại:

$$C^{2753} \equiv (P^{17})^{2753} \equiv P(P^{3120})^{15} \equiv P(\text{mod } 3233)$$

ở đây ta dùng định lí Euler để nhận được $P^{\phi(3253)} \equiv P^{3120} \equiv 1(\text{mod } 3233)$, khi $(P,3233)=1$ (điều này đúng với mọi khối trong thông báo của chúng ta)

Bây giờ ta chỉ ra rằng, hệ mã RSA thoả mãn các nguyên tắc của hệ mã khoá công khai nói ở đầu tiết này. Trước tiên, ta chú ý rằng, mỗi cá thể phải chọn hai số nguyên tố lớn p và q, cỡ chừng 100 chữ số thập phân. Điều này có thể là trong ít phút nhờ một máy tính. Khi các số nguyên tố p và q đã được chọn, số mũ dùng để mã hoá e sẽ được lấy sao cho $(e, \phi(qp))=1$. Nói chung nên chọn e là số nguyên tố tuỳ ý lớn hơn q và p. Số e được chọn nhất thiết phải thoả mãn $2^e>n=pq$. Nếu điều kiện này không được thoả mãn, ta có $C=P^e<n$, và như vậy để tìm ra P, ta chỉ việc tính căn bậc e của C. Khi điều kiện $2^e>n$ được thoả mãn, mọi khối P khác 0 và 1 đều được mã hoá bằng cách nâng lên luỹ thừa và lấy đồng dư theo modulo n.

Ta cần phải chứng tỏ rằng, việc biết khoá lập mã (công khai) (e,n) không đẫn đến việc tìm được khoá giải mã (d,n).

Chú ý rằng, để tìm nghịch đảo d của e modulo ϕ (n), trước tiên phải tìm được ϕ (n). Việc tìm ϕ (n) không dễ hơn so với phân tích n, bởi vì, một khi biết ϕ (n) và n, ta sẽ phân tích được $n=pq$.

Thật vậy, ta có:

$$p+q=n-\phi(n)+1$$

$$p-q=\sqrt{(p+q)^2-4qp}=\sqrt{(p+q)^2-4n}$$

Từ các công thức đó tìm được q và p.

Nếu ta chọn các số p và q khoảng 100 chữ số thập phân, thì n sẽ có khoảng 200 chữ số thập phân. Để phân tích một số nguyên cỡ lớn như thế, với các thuật toán nhanh nhất hiện nay và với những máy tính hiện đại nhất, ta mất khoảng 3,8 tỷ năm!

Có một vài điều cần lưu ý khi chọn các số p và q để tránh rơi vào trường hợp tích pq bị phân tích nhanh nhờ những thuật toán đặc biệt: q và p cần chọn sao cho p-1 và q-1 chỉ có các thừa số nguyên tố lớn, $(p$-1,q-1$)$ phải nhỏ, q và p phải có số chữ số trong khai triển thập phân khác nhau không nhiều.

Có thể nảy ra câu hỏi: trong một hệ thống nhiều cá thể tham gia, các khoá lập mã đã lại được công khai, làm sao có thể tránh được trường hợp một cá thể này "mạo danh" một cá thể khác để gửi thông báo cho một cá thể thứ ba? Nói cách khác làm sao có thể "kí tên" dưới các thông báo mật? Vấn đề này được giải quyết đơn giản như sau: Giả sử "ông I" cần kí tên dưới thông báo gửi "ông J". Khi đó, trước tiên, ông I tính

$$S\equiv D_{k_i}(I)\equiv I^{d_i}\ (\mathrm{mod}\ n_i).$$

Chú ý rằng chỉ có ông I làm được việc này, vì trong công thức sử dụng khoá giải mã của ông I. Sau đó, I sẽ gửi cho J thông báo

$$C\equiv E_{k_j}(S)=S^{e_j}\ (\mathrm{mod}\ n_j),$$

trong đó (e_j,n_j) là khoá lập mã của J.

Khi nhận được, để giả mã, J trước tiên dùng khoá giải mã riêng của mình để nhận ra S:

$$D_{k_j}(C) \equiv D_{k_j}(E_{k_j}(S)) \equiv \mathrm{S}$$

Để xác minh *S* đích thực là chữ kí của *I, J* chỉ còn việc áp dụng vào *S* khoá lập mã công khai của *I:*

$$E_{k_i}(S) \equiv E_{k_i} D_{k_i}(I) \equiv I$$

Chú ý cách là như trên thích hợp khi $n_j > n_i$, vì khi đó ta luôn có $S < n_j$. Nếu ngược lại, *I* phải tách *S* thành từng khối có độ dài bé hơn n_j và mã hoá từng khối rồi mới chuyển.

Như vậy, một mặt *J* xác định được đó đúng là thông báo do *I* gửi đến, mặt khác *I* cũng không thể từ chối việc mình là chủ nhân của thông báo đó, vì ngoài *I* ra, không ai có khoá mã D_{k_i} để mạo "chữ kí" của *I*.

Trên đây là hệ mật mã khoá công khai xuất hiện đầu tiên. Từ đó đến nay, có nhiều hệ mật mã khoá công khai mới ra đời. Tuy vậy, nguyên tắc chung của cac hệ mã đó là sử dụng những "thuật toán một chiều", tức là những thuật toán cho phép tìm ra một đại lượng nào dó tương đối nhanh, nhưng việc tìm "nghịch đảo" (theo một nghĩa nào đó) của nó đòi hỏi thời gian quá lớn. Độc giả nào quan tâm đến vấn đề này có thể tìm đọc trong những tài liệu chuyên về lí thuyết mật mã. Trong chương tiếp theo, ta sẽ quay về với lí thuyết mật mã khoá công khai khi nghiên cứu các đường cong elliptic.

Cùng với sự phát triển của mật mã khoá công khai, có lẽ sẽ đến lúc bên cạnh địa chỉ và điện thoại của mỗi cơ quan, công ty, còn ghi thêm khoá lập mã của họ!

Bài tập và tính toán thực hành chương 6.

I. Bài tập

6.1. Biết rằng thông báo sau đây đã được mã hoá bằng mã Ceasar (với khoá k nào đó trong khoảng 1-29), hãy tìm khoá và giải mã:

SÔEMR IEIEH USSOT SLUƠI EIÔHE ITSAÂ UƠIEI ÔLUƠI ESÔYB SOSÔE MRDEI EIÔXÂ EIÔBS ORMCE SXSÔL GDESÔ MBSOĂ ÔMTMR

6.2. Dùng mã khối để mã hoá câu

CO CÔNG MAI SĂT CO NGAY NÊN KIM

với khoá ma trận là

$$\begin{pmatrix} 24 & 22 \\ 11 & 10 \end{pmatrix}$$

6.3. Giải mã câu sau đây, biết rằng nó được mã hoá bằng khối với ma trận

$$\begin{pmatrix} 8 & 4 \\ 17 & 11 \end{pmatrix}$$

OD OÂ XC ƠƠ EP YƠ NR EY

6.4. Có thể lập mã khối theo cách sau đây. Giả sử A, B là các ma trận vuông cấp hai. Quá trình mã hoá được thực hiện bởi công thức

$$C \equiv AP+B \pmod{29}.$$

Hãy viết công thức giả mã và cho một ví dụ cụ thể.

6.5. Mã hoá câu sau đây bằng mã mũ với p=3137, e=31:

ĐÊN NƠI AN TOAN

6.6. Hãy giải mã văn bản mật sau đây, nếu biết nó được mã hoá bằng mã mũ với p=3137, e=31:

0206 0248 1345 2200

6.7. Chứng minh rằng, khi lập mã RSA, nếu xảy ra trường hợp có một từ P nào đó trong thông báo không nguyên tố cùng nhau với khoá $n=pq$ đã chọn, và từ này bị phát hiện, thì nhân viên phân tích mã có thể phân tích được n ra thừa số nguyên tố, và do đó, tìm được khoá giải mã.

6.8. Chứng minh rằng, nếu các số q, p được chọn đủ lớn thì trường hợp "rủi ro" nói trong bài tập 6.7 xảy ra với xác suất rất nhỏ.

6.9. Dùng khoá với n=3233, e=17 để mã hoá câu

CHUC MƯNG NĂM MỚI

II. Thực hành tính toán trên máy

Để làm giảm nhẹ các thao tác trong việc lập mã và giải mã văn bản đối với mã khối và mã mũ chúng tôi chỉ ra cách dùng Maple để tính toán.

Để thống nhất ta gọi văn bản là thông báo cần chuyển được viết bằng ngôn ngữ thông thường không có dấu, P là chữ trong văn bản. Bản đã mã hoá của văn bản gọi là văn bản mật, C chữ số tương ứng trong văn bản mật. Giải mã tức là chuyển văn bản mật C thành văn bản ban đầu P.

Do trong Maple không có chế độ tiếng Việt, nên ta dùng kí hiệu aw, aa, dd, ee, oo, ow, uw thay cho các chữ ă, â, đ, ê, ô, ơ, ư tương ứng.

II. 1. Thực hành lập mã và giải mã khối

1. Lập mã: Đối với hệ mã khối và mã mũ, ta ứng các chữ trong văn bản với các số, chuyển các số đó thành hệ thống số khác thông qua khoá lập mã, sau đó lại dùng bảng tương ứng các số vừa tìm được ta được văn bản mật cần chuyển.

Giả sử ta cần mã hoá văn bản P bằng mã khối 2 chữ với khoá lập mã là $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$.

Khoá lập mã được dùng ở đây là ma trận cấp 2×2, nên nếu số các chữ trong văn bản P là số chẵn thì việc mã hoá xảy ra bình thường nhưng nếu số các chữ trong văn bản P là số lẻ thì chữ cuối cùng của văn bản P sẽ không được mã hoá. Để khắc phục tình trạng đó trong trường hợp thứ 2 ta thêm vào cuối văn bản P một chữ mà không ảnh hưởng đến nội dung của văn bản (chẳng hạn chữ x). Ta thực hiện theo các bước sau đây:

Bước 1: Tính số các chữ trong văn bản P, ta thực hiện bằng dòng lệnh:

```
[>nops([P]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên số các chữ cái trong văn bản P, nếu đó là số chẵn ta không thay đổi P, nếu đó là số lẻ ta thêm vào cuối văn bản P một chữ cái, ví dụ là chữ x.

Bước 2: Thiết lập tương ứng các chữ cái trong văn bản P (hoặc là P sau khi đã thêm chữ cái x) với các số thông qua dòng lệnh sau đây:

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[P]):
```

Sau dấu (:) ấn phím "Enter", vì ở đây ta không cần hiển thị kết quả của dòng lệnh này nên dùng dấu (:) thay cho dấu (;). Khi đó trên màn hình sẽ hiện lên dấu nhắc ([>) để thực hiện tiếp dòng lệnh thứ 3.

Bước 3: Thực hiện dòng lệnh:

```
[>N:=nops(L)/2:
```

(Lệnh `nops(L)` dùng để tính số phần tử của L)

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 4.

Bước 4: Thực hiện dòng lệnh:

```
[> subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({x-
a1*L[2*k-1]-a2*L[2*k],y-a3*L[2*k-1]-a4*L[2*k]},29),k=1..
N)]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên các chữ tương ứng của văn bản mật. Ta viết lại theo thứ tự thích hợp sẽ được văn bản mật cần chuyển.

Như vậy, rất đơn giản, để mã hoá văn bản nào đó ta chỉ cần thay các chữ cái trong văn bản vào vị trí của P (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh thứ nhất, thứ hai và thay các số a_1a_2,a_3,a_4 của khoá lập mã vào dòng lệnh thứ tư. Để dễ hiểu ta theo dõi thí dụ sau đây:

Chú ý: Đối với hệ mã này ta cho tương ứng chữ *y* với số 0.

Thí dụ 1: Mã hoá câu CHUC BAN THANH CÔNG bằng mã khối với khoá lập mã là $\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>nops([c,h,u,c,b,a,n,t,h,a,n,h,c,oo,n,g]);
                              16
```

16 là một số chẵn do đó ta thực hiện tiếp dòng lệnh thứ hai mà không cần phải thêm chữ vào.

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[c,h,u,c,b,a,n,t,h,
a,n,h,c,oo,n,g]):

[> N:=nops(L)/2:

[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y},   [seq   (msolve   ({x-
23*L[2*k-1]-11*L[2*k],y-9*L[2*k-1]-
12*L[2*k]},29),k=1..N)]);


[{y = aa, x = b},{y = t, x = q},{y = ow, x = n},{y = uw,
x = s},{y = t, x = aa},{x = u, y = m},{y = y, x = s},{x
= l, y = aa}]
```

Vậy ta có văn bản mật tương ứng là BÂ QT NƠ SƯ ÂT UM SY LÂ

Thí dụ 2: Mã hoá câu LY THUYÊT MÂT MA KHÔNG CO GI KHO bằng mã khối với khoá lập mã là $\begin{pmatrix} 8 & 4 \\ 17 & 11 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>nops([l,y,t,h,u,y,ee,t,m,aa,t,m,a,k,h,oo,n,g,c,o,g,i,k
,h,o]);
                                25
```

25 là một số lẻ do đó ta phải thêm một chữ x vào trong văn bản P.

```
[>L:=subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,
h=11,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21
,r=22,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[l,y,t,h,u,y,e
e,t,m,aa,t,m,a,k,h,oo,n,g,c,o,g,i,k,h,o,x]):

[> N:=nops(L)/2:

[>subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=
h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22
=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},[seq (msolve ({x-
8*L[2*k]-4*L[2*k-1],y-17*L[2*k]-11*L[2*k-
1]},29),k=1..N)]);


[{x = v, y = ee}, {x = g, y = n}, {x = k, y = l}, {x =
u, y = l},{x = uw, y = k}, {x = k, y = uw}, {x = q, y =
y}, {x = l, y = q},{y = v, x = x}, {x = h, y = u}, {x =
p, y = t}, {y = h, x = t},{x = aw, y = u}]
```

Vậy ta có văn bản mật tương ứng là VÊ GN KL UL ƯK KƯ QY LQ XV HU PT TH ĂU

2. Giải mã: Giả sử ta nhận được văn bản mật C, cần giải mã C với khoá ma trận A=$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ ta thực hiện trình tự từng bước như sau:

Bước 1: Lập tương ứng mỗi chữ cái trong văn bản mật với một số bằng dòng lệnh như quá trình lập mã.

```
[>L:=
subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=11,
i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=22
,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[C]):
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 2.

Bước 2: Thực hiện dòng lệnh:

```
[>N:=nops(L)/2:
```

(Lệnh `nops(L)` dùng để tính số phần tử của L)

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ xuất hiện dấu nhắc lệnh ([>), ta thực hiện tiếp bước 3.

Bước 3: Thực hiện dòng lệnh:

```
[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({L[2*k-
1]-a1*x-a2*y,L[2*k]-a3*x-a4*y},29),k=1..N)]);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên các chữ tương ứng của văn bản. Ta viết lại theo thứ tự thích hợp sẽ được văn bản ban đầu.

Như vậy, rất đơn giản, để giải mã văn bản mật C khi biết khoá ma trận A ta chỉ cần thay các chữ cái trong văn bản mật vào vị trí của C (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh thứ nhất và thay các số a_1a_2,a_3,a_4 của khoá lập mã vào dòng lệnh. Để dễ hiểu ta theo dõi thí dụ sau đây:

Thí dụ: Giải mã văn bản mật BÂ QT NƠ SƯ ÂT UM SY LÂ bằng mã khối với khoá lập mã là $\begin{pmatrix} 23 & 11 \\ 9 & 12 \end{pmatrix}$.

Ta thực hiện như sau:

```
[>L:=subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,
h=11,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21
,r=22,s=23,t=24,u=25,uw=26,v=27,x=28,y=0},[b,aa,q,t,n,ow
,s,uw,aa,t,u,m,s,y,l,aa]):

[> N:=nops(L)/2:

[>subs
({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i
,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=
s,24=t,25=u,26=uw,27=v,28=x,0=y}, [seq (msolve ({L[2*k-
1]-23*x-11*y,L[2*k]-9*x-12*y},29),k=1..N)]);


[{y = h, x = c}, {y = c, x = u}, {y = a, x = b}, {y = t,
x = n},{x = h, y = a}, {y = h, x = n}, {y = oo, x = c},
{y = g, x = n}]
```

Như vậy ta có văn bản là CHUC BAN THANH CÔNG.

II. 2. Thực hành lập mã và giải mã mũ

1. Lập mã: Đối với hệ mã mũ, ta ứng các chữ trong văn bản với các số, chuyển các số đó thành hệ thống số khác thông qua khoá lập mã, sau đó lại dùng bảng tương ứng các số vừa tìm được ta được văn bản mật cần chuyển. Giả sử *p* là một số nguyên tố lẻ (để đảm bảo tính an toàn số *p* được chọn ở đây phải là số nguyên tố tương đối

lớn, chẳng hạn lớn hơn 2929), *e* là khoá lập mã (trong đó (e,p-1)=1). Để chuyển một văn bản cho đối tượng có khoá lập mã là (e,p) tiến hành lập mã theo các bước sau:

Bước 1: Tìm *m* (là số nguyên lớn nhất sao cho mọi số tương ứng với *m* chữ cái đều nhỏ hơn *p)*. Ta thực hiện dòng lệnh sau:

```
[>Lp:=length(p);
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên kết quả.

Nếu kết quả là một số lẻ thì lấy *m* =(*Lp*-1)/2.

Nếu kết quả là một số chẵn xét p'=2929...29 trong đó số chữ số của p' bằng số chữ số của *p,* ta thực hiện tiếp dòng lệnh:

```
[>p-p';
```

Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên kết quả. Nếu kết quả là một số dương lấy *m=Lp/2*. Nếu kết quả là một số âm lấy *m=(Lp-2)/2*.

Bước 2: Đặt tương ứng mỗi chữ trong văn bản *P* với một số, ta dùng dòng lệnh sau:

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[P]);
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả.

Bước 3: Chia các số tương ứng tìm được thành từng nhóm *m* số, trong đó các số 1,2,3,4,5,6,7,8,9 thay bởi 01,02,03,04,05,06,07,08,09. Nếu nhóm cuối cùng chưa đủ m số thì ta thêm vào các số 28 (tương ứng với chữ x). Công việc này ta thực hiện bằng tay vì nó đơn giản. Rồi tìm các chữ tương ứng trong văn bản mật. Ta thực hiện dòng lệnh:

```
[>L:=[nhóm thứ nhất, nhóm thứ hai,...]:seq (msolve (x-
L[k]&^e,p),k=1..nops(L));
```

Sau dấu (:) ấn phím "Enter" trên màn hình sẽ hiện ra kết quả.

Như vậy, rất đơn giản, để mã hoá văn bản nào đó ta chỉ cần thay các chữ cái trong văn bản vào vị trí của P (với chú ý các chữ cái phải tách biệt nhau bởi dấu (,)) trong dòng lệnh ở bước 2, và thay các số *p,e* của khoá lập mã vào các dòng lệnh ở bước 1, bước 3. Để dễ hiểu ta theo dõi thí dụ sau đây:

Thí dụ: Với khoá lập mã là p=2938,e=31. Mã hoá thông báo sau:

ĐI HA NÔI NGAY

Ta thực hiện như sau:

```
[>length(2839);
```

```
                                4
```

4 là một số chẵn do đó ta phải thực hiện tiếp lệnh thử với p'=2929

```
[> 2939-2929;
```

```
                                10
```

10 là một số dương do đó lấy m=4/2=2.

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[dd,i,h,a,n,oo,i
,n,g,a,y]);

              [7, 12, 11, 1, 16, 18, 12, 16, 10, 1, 29]

[>          L:=[0712,1101,1618,1216,1001,2928]:seq(msolve
(L[k]&^31-x,2939),k=1..nops(L));


{x = 898}, {x = 1853}, {x = 1003}, {x = 2156}, {x =
1786}, {x = 2614}
```

Vậy ta có văn bản mật là 898 1853 1003 2156 1786 2614.

2. Giải mã: Khi nhận được một văn bản mật C gửi cho mình, ta dùng khoá giải mã *(d,n)* của mình để tìm ra nội dung nhận được. Ta thực hiện các dòng lệnh như sau:

Bước 1: Tìm lại khối P' (tương ứng bằng các số) trong văn bản, ta dùng dòng lệnh:

```
[>L:=[C]: seq (msolve (x-L[k]&^d,p),k=1..nops(L));
```

Chú ý khi thay C vào trong dòng lệnh này thì các khối phải cách nhau bởi (,). Sau dấu (;) ấn phím "Enter" trên màn hình sẽ hiện lên khối các số tương ứng của P'.

Bước 2: Để nhận được P ta tách mỗi khối của P' nhận được thành các nhóm có hai số rồi tương ứng mỗi nhóm với một chữ cái. Ta thực hiện như sau:

```
[>P:=[P']:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,
10=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p
,21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,29=y},[seq((P[
i] mod  29),i=1..nops(P))]);
```

Ta xét thí dụ sau:

Thí dụ: Hãy giải mã văn bản mật 898 1853 1003 2156 1786 2614 biết khoá giải mã là (853,2939).

Ta thực hiện như sau:

```
[>L:=[898,1853,1003,2156,1786,2614]:seq(msolve(x-
L[k]&^853,2939),k=1..nops(L));


{x = 712}, {x = 1101}, {x = 1618}, {x = 1216}, {x =
1001}, {x = 2928}


[>P:=
[07,12,11,1,16,18,12,16,10,1,29,28]:subs({1=a,2=aw,3=aa,
4=b,5=c,6=d,7=dd,8=e,9=ee,10=g,11=h,12=i,13=k,14=l,15=m,
```

```
16=n,17=o,18=oo,19=ow,20=p,21=q,22=r,23=s,24=t,25=u,26=u
w,27=v,28=x,0=y},[seq((P[i] mod  29),i=1..nops(P))]);


                [dd, i, h, a, n, oo, i, n, g, a, y, x]
```

Vậy văn bản nhận được là ĐI HA NÔI NGAY.

II. 3. Thực hành lập mã và giải mã RSA

Quá trình này được thực hiện tương tự đối với hệ mã mũ, ta chỉ cần thêm vào chữ kí của mình. Đối với hệ mã này, khoá lập mã là (e,n) trong đó *n* là tích của hai số nguyên tố lớn, khoá giải mã là (e,d). Ta xét thí dụ sau đây:

1. Lập mã:

Thí dụ : Linh có khoá lập mã là (19,221), Lan có khoá lập mã là (13,1457). Linh muốn gửi cho Lan lời nhắn sau: "Anh muốn gặp em, Linh". Anh ta thực hiện như sau:

Bước 1: Ứng mỗi chữ trong lời nhắn với một số, thực hiện bằng dòng lệnh:

```
[>subs({a=1,aw=2,aa=3,b=4,c=5,d=6,dd=7,e=8,ee=9,g=10,h=1
1,i=12,k=13,l=14,m=15,n=16,o=17,oo=18,ow=19,p=20,q=21,r=
22,s=23,t=24,u=25,uw=26,v=27,x=28,y=29},[a,n,h,m,u,oo,n,
g,aw,p,e,m,l,i,n,h]);
```

Sau khi ấn phím "Enter" ta nhận được kết quả:

```
    [1, 16, 11, 15, 25, 18, 16, 10, 2, 20, 8, 15, 14,
12, 16, 11]
```

Bước 2: Linh kí tên của mình, trong quá trình này Linh dùng đến khoá giải mã của mình là (91,221)

```
[>L:=[14,12,16,11]:seq(msolve(L[k]&^91-
x,221),k=1..nops(L));


              {x = 27}, {x = 142}, {x = 16}, {x = 80}
```

Bước 3: Thực hiện mã hoá các số nhận được, kể cả chữ kí của Linh

```
[>L:=[1,16,11,15,25,18,16,10,2,20,8,15,14,12,16,11,27,14
2,16,80]:seq(msolve (L[k]&^13-x,1457),k=1..nops(L));


{x = 1}, {x = 252}, {x = 207}, {x = 1360}, {x = 862}, {x
= 237},{x = 252}, {x = 226}, {x = 907}, {x = 1002}, {x =
1287}, {x = 1360}, {x = 679}, {x = 1040}, {x = 252}, {x
= 207},{x = 1207}, {x = 330}, {x = 252}, {x = 919}
```

Vậy Linh sẽ gửi cho Lan lời nhắn 1 252 207 1360 862 237 252 226 907 1002 1287 1360 679 1040 252 207 1207 330 252 919

2. Giải mã:

Thí dụ : Khi nhận được lời nhắn Lan sẽ giải mã theo các bước sau: (khoá giải mã của Lan là (637,1457):

```
[>L:=[1,252,207,1360,862,237,252,226,907,1002,1287,1360,
679,1040, 252, 207,1207,330,252,919]:seq(msolve
(L[k]&^637-x,1457),k=1..nops(L));


{x = 1}, {x = 16}, {x = 11}, {x = 15}, {x = 25}, {x =
18}, {x = 16},{x = 10}, {x = 2}, {x = 20}, {x = 8}, {x =
15}, {x = 14},{x = 12}, {x = 16}, {x = 11}, {x = 27}, {x
= 142}, {x = 16},{x = 80}


[>P:=[1,16,11,15,25,18,16,10,2,20,8,15,14,12,16,11,27,14
2,16,80]:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,8=e,9=ee,1
0=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,19=ow,20=p,
21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},[seq((P[i]
mod  29),i=1..nops(P))]);


   [a, n, h, m, u, oo, n, g, aw, p, e, m, l, i, n, h, v,
uw, n, r]
```

Lan giải ra được lời nhắn là "Anh muốn gặp em Linh v ư n r", như vậy người nhắn là Linh và 4 chữ cuối là chữ kí của Linh. Để kiểm tra xem có đúng thật hay không, Lan thực hiện tiếp các dòng lệnh:

```
[>L:=[27,142,16,80]:seq(msolve(L[k]&^19-
x,221),k=1..nops(L));


                {x = 14}, {x = 12}, {x = 16}, {x = 11}


[>P:=[14,12,16,11]:subs({1=a,2=aw,3=aa,4=b,5=c,6=d,7=dd,
8=e,9=ee,10=g,11=h,12=i,13=k,14=l,15=m,16=n,17=o,18=oo,1
9=ow,20=p,21=q,22=r,23=s,24=t,25=u,26=uw,27=v,28=x,0=y},
[seq((P[i] mod  29),i=1..nops(P))]);


                                [l, i, n, h]
```

Vậy người gửi đúng là Linh.

Chương 7

ĐƯỜNG CONG ELLIPTIC

§1 Định nghĩa.

Chương này nhằm trình bày những khái niệm cơ bản của một đối tượng rất quan trọng của lí thuyết số và hình học đại số: các đường cong elliptic. Về mặt lịch sử, các đường cong elliptic xuất hiện lần đầu tiên trong các nghiên cứu về tích phân elliptic (từ đó có tên gọi của đường cong). Các đường cong này có mặt trong nhiều lĩnh vực khác nhau của toán học vì nó rất phong phú về mặt cấu trúc. Một mặt, đó là đường cong không kì dị, tức là các đa tạp một chiều. Mặt khác, các điểm của đương cong lập thành một nhóm Aben. Vì thế hầu như mọi công cụ của toán học đều được áp dụng vào nghiên cứu đường cong elliptic. Ngược lại, những kết quả về đường cong elliptic có ý nghĩa quan trọng đối với nhiều vấn đề khác nhau. Xin chỉ ra một vài ví dụ. Về mặt lí thuyết, định lí lớn Fermat đã được chứng minh (trong công trình của A. Wiles) bằng cách chứng minh giả thuyết Taniyama-Weil về các đường cong elliptic. Về mặt ứng dụng, rất gần đây, các đường cong elliptic được dùng trong việc xây dựng một số hệ mật mã khoá công khai.

Để có thể trình bày tương đối sâu về đường cong elliptic, chúng ta cần nhiều hiểu biết về hình học đại số. Bởi vậy, chúng tôi chỉ có thể đề cập ở đây những khái niệm cơ bản nhất. Mục đích của chương chỉ là làm thế nào để độc giả có thể hình dung lí do tại sao đường cong elliptic lại có nhiều ứng dụng như vậy. Mặt khác, chúng tôi cũng giới thiệu sơ lược một vài thuật toán liên quan đến đường cong elliptic trên trường hữu hạn. Trong khi trình bày, cũng giống như các phần khác của cuốn sách, chúng tôi luôn cố gắng dùng ngôn ngữ "sơ cấp" nhất có thể. Bởi vậy, đôi khi phải bỏ qua chứng minh. Độc giả nào quan tâm sâu hơn về các đường cong elliptic, có thể tìm đọc trong các tài liệu [Ha], [Sil].

Định nghĩa 7.1. *Đường cong elliptic trên trường K* là tập hợp các điểm *(x,y)* thoả mãn phương trình

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6, \qquad (7.1)$$

với một điểm O gọi là *điểm tại vô cùng (*sẽ nói rõ về sau). Hơn nữa, phương trình (7.1) phải thoả mãn điều kiện *không kì dị*, tức là, nếu viết nó dưới dạng *F(x,y)=0* thì tại mọi điểm *(x,y)* thoả mãn phương trình, có ít nhất một trong các đạo hàm riêng $\partial F/\partial x, \partial F/\partial y$ khác 0.

Điều kiện không kì dị nói trên tương đươmg với điều kiện, nếu xét tập hợp các điểm nói trên như một đường cong, thì đường cong đó không có điểm bội. Như vậy, nếu biểu diễn y^2 như là một đa thức bậc 3 của x, thì đa thức đó không có nghiệm bội.

Chú ý rằng, phương trình trên đây không duy nhất: trong nhiều trường K, có thể tìm được "dạng tối thiểu" của phương trình biểu diễn đường cong.

Nếu ta xét phương trình (7.1) với các hệ số trong Z, thì vì Z có thể nhúng vào trong mọi trường K tuỳ ý nên có thể xét phương trình trên như là phương trình trong trường K. Một điều cần lưu ý ngay: phương trình đó có thể thoả mãn điều kiện không kì dị đối với trường này, nhưng lại không thoả mãn điều kiện đó đối với trường khác. Chẳng hạn, nếu trường đang xét có đặc trưng 2 thì ta có $(x^2)'=0$ với mọi x!

Điểm tại vô cùng nói trong định nghĩa là điểm vô cùng trong đường cong xạ ảnh tương ứng. Ta xét không gian xạ ảnh P^2, tức là không gian mà các điểm là các lớp tương đương của các bộ ba (x,y,z), trong đó x, y, z không đồng thời bằng 0, và bộ ba (x,y,z) tương đương với bộ ba $(\lambda x, \lambda y, \lambda z)$, $\lambda \neq 0$. Như vậy, nếu $z \neq 0$ thì lớp tương đương của (x,y,z) chứa bộ ba $(x/z,y/z,1)$. Ta có thể đồng nhất mặt phẳng xạ ảnh P^2 với mặt phẳng thông thường (aphin) cùng với các "điểm tại vô hạn" ứng với $z=0$.

Một đường cong trong mặt phẳng thông thường có thể tương ứng với đường cong trong mặt phẳng xạ ảnh bằng cách thêm vào các điểm tại vô cùng. Để làm việc đó, trong phương trình xác định đường cong, ta chỉ cần thay x bởi x/z, y bởi y/z và nhân hai vế của phương trình với một luỹ thừa thích hợp của z để khử mẫu số.

Ví dụ. Đường cong elliptic với phương trình (7.1) được thêm vào các điểm tại vô cùng để có đường cong tương ứng trong không gian xạ ảnh:

$$y^2z+a_1xyz+a_3yz^2=x^3+a_2x^2z+a_4xz^2+a_6z^3, \qquad (7.2)$$

Định lí sau đây cho ta thấy có thể định nghĩa phép cộng các điểm trên đường cong elliptic để trang bị cho nó cấu trúc nhóm Aben.

Định lí 7.2. *Xét đường cong elliptic xác định trên trường tuỳ ý bởi phương trình*

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6, \qquad (7.3)$$

Ta có thể trang bị cho tập hợp các điểm của đường cong cấu trúc nhóm Aben cộng tính như sau:

-Phần tử 0 là điểm tại vô cùng; (0,1,0).

-Điểm với toạ độ (x_1,y_1) có nghịch đảo là điểm với toạ độ($x_1,-y_1-a_1x_1-a_3$).

- Nếu hai điểm $P_1=(x_1,y_1)$ và $P_2 = (x_2,y_2)$ không phải là nghịch đảo của nhau thì $P_1+P_2=P_3$, $P_3 =(x_3,y_3)$xác đinh như sau.

Đặt

$$m=\frac{y_1-y_2}{x_1-x_2}, \text{ nếu } P_1 \neq P_2 \ ;$$

$$m=\frac{3x_1^2+2a_2x_1+a_4-a_1x_1}{2y_1+a_1x_1+a_3}, \text{ nếu } P_1=P_2.$$

và tính x_3,y_3 theo công thức

$$x_3=-x_1-x_2-a_2+m\,(m+a_1),$$

$$y_3=-y_1-a_3-a_1x_3+m\,(x_1-x_2)$$

Chứng minh. Bằng tính toán trực tiếp dựa vào phương trình xác định đường cong, dễ kiểm tra định nghĩa phép cộng trên đây thoả mãn các tiên đề của nhóm Aben.

Để thấy rõ ý nghĩa hình học của định nghĩa phép cộng trên đây, ta xét trường hợp quan trọng sau đây của các đường cong elliptic trên trường thực *R*.

§2. Đường cong elliptic trên trường thực. Trước tiên, ta có nhận xét sau đây. Trong những trường với đặc trưng khác 2 và 3, phương trình (7.1) có thể đưa về dạng

$$Y^2=4X^3+c_4X+c_6. \qquad (7.4)$$

Thật vậy, chỉ cần dùng phép đổi biến:

$$Y=2y+a_1x+a_3$$

$$X=x+(a_1^2+4a_2)/12$$

Để đơn giản, ta thường dùng dạng sau đây, gọi là *dạng Weierstrass* của đường cong:

$$y^2=x^3+a_4x+a_6.$$

Trong trường hợp này, biệt thức Δ của đường cong là

$$\Delta =-16(4a_4^3+27a_6^2)$$

Như vậy, điều kiện để đường cong không có kì dị (không có điểm bội)là:

$$4a_4^3+27a_6^2 \neq 0.$$

Ta sẽ sử dụng dạng Weierstrass của đường cong. Bằng tính toán trực tiếp tọa độ các điểm theo công thức đã cho trong định lí 7.2, ta có thể thấy luật cộng trong nhóm lập bởi các điểm của đường cong có mô tả hình học sau đây:

Nếu các điểm *P* và *Q* của đường cong có toạ độ *x* khác khác nhau thì đường thẳng đi qua *P* và *Q* sẽ cắt đường cong tại một điểm thứ ba. Điểm đối xứng với giao điểm đó qua trục hoành chính là diểm *P+Q*.

Trong trường hợp *P* và *Q* có cùng hoành độ, tung độ của chúng sẽ là các giá trị đối nhau, và *P,Q* là hai điểm đối xứng nhau qua trục hoành. Khi đó đường thẳng đi qua *P,Q* sẽ "cắt" đường cong tại vô cùng: đó chính là điểm 0 của nhóm cộng các điểm, và P,Q là các phần tử nghịch đảo của nhau.

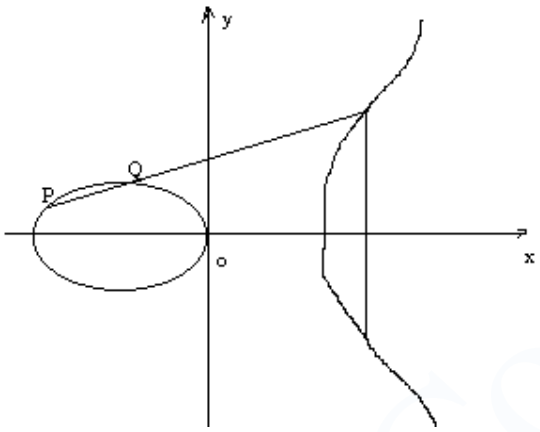
Rõ ràng "cộng" *P* với 0, thực hiện bằng cách nối *P* với điểm tại vô cùng bằng đường thẳng song song với trục tung sẽ cắt đường cong tại điểm đối xứng với *P* qua trục hoành, và như vậy *P+0=P*.

Trên hình 1 ta minh hoạ những điều vừa nói trên qua ví dụ đường cong với phương trình $y^2=x^3-x$.

Vì các điểm của đường cong là các phần tử của một nhóm cộng Aben, ta sẽ dùng kí hiệu *NP* để chỉ phần tử nhận được bằng cách cộng liên tiếp *N* lần điểm *P*.

Định nghĩa 7.3. Điểm *P* của đường cong được gọi là *điểm bậc hữu hạn* nếu tồn tại số nguyên dương *N* sao cho *NP=0*. Số *N* nhỏ nhất thoả mãn điều kiện đó gọi là bậc của *P*.

Dĩ nhiên không phải mọi điểm của đường cong đều có bậc hữu hạn.



Hình 1. Đường cong elliptic $y^2=x^3-x$ trên trường thực

§3. Đường cong elliptic trên trường các số hữu tỷ.

Trong rất nhiều vấn đề của Hình học đại số và số học, ta thường phải làm việc với các đường cong trên trường số hữu tỷ. Đó là các đường cong cho bởi phương trình (7.2), trong đó các hệ số là các số hữu tỷ, và ta cũng chỉ xét các điểm với toạ độ là các số hữu tỷ. Nghiên cứu đường cong elliptic trên trường số hữu tỷ cũng có nghĩa là nghiên cứu tập hợp nghiệm hữu tỷ của phương trình (7.2), một vấn đề quan trọng của số học. Trong phần cuối chương, ta sẽ thấy rằng, vấn đề này còn liên quan đến chứng minh định lí lớn Fermat.

Giả sử *E* là đường cong elliptic đã cho. Ta kí hiệu qua *E(Q)* tập hợp các điểm có toạ độ hữu tỷ. Như ta đã thấy, tập hợp này có cấu trúc nhóm Aben. Các điểm bậc hữu hạn của nhóm Aben *E(Q)* lập thành nhóm con $E(Q)_{tors}$, gọi là *nhóm con xoắn* của *E(Q)*. Khi đó, *E(Q)* sẽ là tổng trực tiếp của $E(Q)_{tors}$ với nhóm con các điểm bậc vô hạn. Định lí nổi tiếng của Mordell nói rằng nhóm con các điểm bậc vô hạn chỉ có hữu hạn phần tử sinh, và do đó đẳng cấu với nhóm Z^r, trong đó *r* là một số nguyên không âm. Số *r* gọi là hạng của đường cong, và là một đặc trưng hết sức quan trọng, chứa nhiều thông tin số học về đường cong. Chứng minh các kết luận này đòi hỏi phải sử dụng nhiều kiến thức sâu sắc về hình học đại số,và do đó vượt ra ngoài khuôn khổ của cuốn sách. Ta hạn chế ở đây phát biểu của định lí Mordell.

Định lí (Mordell). *Giả sử E là một đường cong elliptic trên Q. Khi đó tập hợp các điểm của E với toạ độ hữu tỷ E(Q) là một nhóm Aben hữu hạn sinh. Nói cách khác, ta có:*

$$E(Q)= E(Q)_{tors} \oplus Z^r,$$

trong đó r là một số nguyên không âm.

Nhóm con xoắn các điểm bậc hữu hạn của một đường cong có thể tính được không khó khăn lắm, trong khi hạng r lại hết sức khó xác định. Thậm chí, ngay đối với một đường cong cụ thể, chỉ ra r bằng 0 hay khác 0 cũng là một điều hết sức khó khăn. Ta có thể thấy ngay rằng, nếu $r=0$ thì đường cong đang xét chỉ có hữu hạn điểm hữu tỷ, trong trường hợp $r \neq 0$, tồn tại vô hạn điểm hữu tỷ trên đường cong. Điều đó tương đương với việc phương trình đã cho có hữu hạn hay vô hạn nghiệm hữu tỷ, một bài toán khó của số học.

Trong §5, ta sẽ thấy rằng, bài toán tìm điểm hữu tỷ của đường cong elliptic liên quan đến việc thành lập những hệ mật mã kiểu mới, cũng như các thuật toán khai triển nhanh số nguyên cho trước thành thừa số nguyên tố. Đó là những ứng dụng gần đây nhất của lí thuyết đường cong elliptic vào các vấn đề thực tiễn.

Như đã nói ở trên, việc xác định nhóm con xoắn của đường cong elliptic không phải là khó khăn. Tuy nhiên, việc chỉ ra tất cả các khả năng của các nhóm con đó (chỉ tồn tại 15 khả năng khác nhau) lại là một bài toán khó, và mới được giải quyết năm 1977 bằng định lí nổi tiếng sau đây của B. Mazur.

Định lí Mazur. *Giả sử E là đường cong elliptic trên trường Q. Khi đó nhóm con xoắn của $E(Q)$ đẳng cấu với một trong 15 nhóm sau đây :*

$$Z/mZ, \text{ trong đó } 1 \leq m \leq 10, \text{ hoặc } m=12.$$

$$Z/2Z \times Z/2mZ, \text{ với } 1 \leq n \leq 4$$

Như vậy, nhóm xoắn của đường cong elliptic có không quá 16 phần tử.

§4. Đường cong elliptic trên trường hữu hạn.

Để chứng minh một phương trình (hệ số nguyên) nào đó không có nghiệm nguyên, một trong những phương pháp thường được dùng là như sau. Ta xét phương trình mới, nhận được từ phương trình đã cho bằng cách thay các hệ số của nó bởi các thặng dư modulo một số p nào đó. Nếu phương trình này không có nghiệm (đồng dư modulo p) thì phương trình xuất phát cũng không có nghiệm. Việc làm đó được gọi là *sửa theo modulo p*. Rõ ràng rằng phương trình mới đơn giản hơn phương trình đã cho, hơn nữa, để xét nghiệm đồng dư modulo p, ta chỉ cần thử với hữu hạn giá trị. Nếu phương trình đã cho quả thật vô nghiệm, thì trong trường hợp chọn số p một cách may mắn, ta có thể đi đến kết luận đó khá dễ dàng.

Khi nghiên cứu các đường cong elliptic, đặc biệt là các đường cong trên trường số hữu tỷ, người ta cũng thường dùng phương pháp tương tự: sửa theo modulo p. Việc làm đó dẫn đến các đường cong trên trường hữu hạn.

Ta cần lưu ý ngay một điều. Khi "sửa" một đường cong elliptic bằng cách chuyển các hệ số thành các đồng dư modulo p, ta có thể nhận được một đường cong có kì dị. Thật vậy, biệt thức của đường cong (khác không) có thể đồng dư 0 modulo p, và khi đó, đường cong nhận được có điểm bội trên trường hữu hạn. Tuy nhiên, rõ ràng điều đó chỉ xảy ra khi p là một ước số của biệt thức của đường cong xuất phát, và do đó, chỉ xảy ra với một số hữu hạn giá trị của p. Ta nói đường cong elliptic đã cho có *sửa xấu* tại những giá trị của p đó, và có *sửa tốt* tại những giá trị p khác.

Điều cần quan tâm đầu tiên khi nghiên cứu một đường cong elliptic trên trường hữu hạn là: đường cong đó có bao nhiêu điểm? Giả sử E là đường cong elliptic trên trường F_q có q phần tử. Các điểm của đường cong là các cặp (x,y), $x, y \in F_q$ thoả mãn phương trình trong F_q:

$$y^2=x^3+a_4x+a_6$$

Như vậy, nếu với giá trị x, $x^3+a_4x+a_6$ là một thặng dư bình phương modulo q thì sẽ có hai điểm (x,y) và $(x,-y)$ thuộc đường cong. Trong trường hợp ngược lại, không có điểm nào của đường cong ứng với giá trị x. Từ đó, khi q là số nguyên tố, theo định nghĩa của kí hiệu Legendre, số điểm của đường cong ứng với giá trị x là

$$1+\left[\frac{x^3+a_4x+a_6}{q}\right]$$

Thêm điểm tại vô cùng, ta có công thức tính số điểm của đường cong trong trường hợp q là số nguyên tố:

$$\#E(F_q)=1+\sum_{x\in F_q}1+\left[\frac{x^3+a_4x+a_6}{q}\right]$$

Trong trường hợp q không phải là số nguyên tố, trong công thức trên đây, thay cho kí hiệu Legedre, ta hiểu đó là kí hiệu Jacobi, và dấu đẳng thức được thay thế bởi bất đẳng thức $\leq$.

Định lí trên đây cho ta một ước lượng của số điểm của đường cong E trên trường F_q.

Định lí Hasse. *Giả sử N là số điểm của đường cong elliptic xác định trên trường F_q. Khi đó ta có:*

$$|N-(q+1)| \leq 2\sqrt{q}\,.$$

Bạn đọc có thể tìm thấy chứng minh của định lí này trong [Sil].

Một trong những ứng dụng mới nhất của đường cong elliptic trên trường hữu hạn, xuất hiện trong những năm gần đây, là các hệ mật mã khoá công khai elliptic. Phần tiếp theo được dành để trình bày vấn đề đó.

§5. Đường cong elliptic và hệ mật mã khoá công khai.

5.1. Hệ mật mã khoá công khai sử dụng đường cong elliptic dựa trên độ phức tạp của thuật toán tìm số nguyên x sao cho $xB=P$, trong đó P, B là các điểm cho trước của đường cong (nếu số như thế tồn tại). Chú ý rằng, các điểm của đường cong lập thành một nhóm, và ta có thể quan niệm xB như là "B^x": bài toán này hoàn toàn tương tự như bài toán tìm logarit cơ sở b của một số p cho trước (xem chương 6).

Trước tiên, ta cần xét thuật toán tìm bội của một điểm trên đường cong.

Định lí 7.4. *Cho E là một đường cong elliptic trên trường hữu hạn F_q, P là một điểm của đường cong. Khi đó có thể tính toạ độ của điểm kP bằng O(log klog^3 q) phép tính bit.*

Trước khi đi vào chứng minh định lí, ta tìm hiểu sơ qua phương pháp rất thông thường để tìm bội của các điểm trên đường cong: phương pháp nhân đôi liên tiếp. Xét ví dụ sau: giả sử cần tính *205P*. Ta viết :

$$205P=2(2(2(2(2(2P+P)+P)+P)+P))+P)$$

Như vậy, việc tính *205P* được đưa về 4 phép cộng hai điểm của đường cong và 7 phép nhân đôi một điểm cho trước.

Ta giả thiết rằng, trường F_q có đặc trưng khác 2, 3. Trong trường hợp $q=2^r$ hoặc $a=3^r$, có những thuật toán nhanh hơn để tính toạ độ của các bội của một điểm cho trước. Như vậy, phương trình xác định đường cong có thể cho dưới dạng Weierstrass:

$$y^2=x^3+ax+b.$$

Khi đó, theo định lí 7.2, tổng $P+Q=(x_3,y_3)$ của hai điểm khác nhau $P=(x_1,y_1)$ và $Q=(x_2,y_2)$ được tính theo công thức sau:

$$x_3=\left(\frac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2, \tag{7.6}$$

$$y_3=-y_1+\left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3). \tag{7.7}$$

Trong trường hợp $P=Q$, ta có công thức để tính *2P:*

$$x_3=\left(\frac{3x_1^2+a}{2y_1}\right)^2-2x_1, \tag{7.8}$$

$$y_3=-y_1+\left(\frac{3x_1^2+a}{2y_1}\right)(x_1-x_3). \tag{7.9}$$

Như vậy, ta phải dùng không quá 20 phép nhân, chia, cộng, trừ để tính toạ độ của tổng hai điểm khi biết các toạ độ của các điểm đó. Số các phép tính bit đòi hỏi là $O(\log^3 q)$ (xem chương 5). Khi dùng phương pháp nhân đôi liên tiếp, ta phải thực hiện $O(\log k)$ phép tính cộng hai diểm hoặc nhân đôi một điểm (xem chương 5). Như vậy, toàn bộ số phép tính bit phải dùng là $O(\log k \log^3 q)$. Định lí được chứng minh.

Tóm lại, ta có thuật toán thời gian đa thức để tính bội của một điểm. Ngược lại, khi biết *kP* và *P*, việc tìm ra *k* với những thuật toán nhanh nhất hiện nay lại đòi hỏi thời gian mũ. Điều này hoàn toàn tương tự như trong trường hợp các số mũ modulo *p,* và sẽ là cơ sở cho việc xây dựng hệ khoá công khai sử dụng đường cong elliptic.

5.2. Mã hoá nhờ các điểm của đường cong elliptic trên trường hữu hạn.

5.2.1. Như đã thấy trong chương 6, việc chuyển thông báo mật thực hiện bằng cách chuyển nó thành dạng chữ số, mã hoá thông báo "chữ số " này và chuyển đi. Vì thế, để đơn giản khi trình bày, ta sẽ xem thông báo cần chuyển là một số nguyên dương m nào đó.

Việc đầu tiên là phải chọn một đường cong elliptic E nào đó trên trường hữu hạn F_q. Sau đó, phải tìm cách tương ứng số nguyên m với một điểm của đường cong E.

Để dễ hiểu quá trình lập mã, ta sẽ xem đường cong E đã được chọn. Việc chọn đường cong sẽ được trình bày ở tiết sau.

5.2.2. Tương ứng một số m với một điểm của đường cong elliptic.

Cho đến nay, chưa có một thuật toán *quyết định* nào hữu hiệu để tìm được một số đủ lớn các điểm của đường cong elliptic. Thuật toán mà ta trình bày sau đây là một thuật toán xác suất với thời gian đa thức.

Trước hết ta chọn một số k nào đó theo yêu cầu sau: trường hợp thuật toán sẽ tiến hành không cho kết quả mong muốn chỉ xảy ra với xác suất không vượt quá 2^{-k}. Như vậy, nói chung $k=40$ là có thể chấp nhận được (ta nhắc lại rằng, trong trường hợp đó, xác suất sai lầm của một thuật toán sẽ bé hơn xác suất sai lầm của phần cứng của máy tính).

Giả sử số m nằm trong khoảng $1 \le m \le M$. Ta luôn chọn q sao cho $q>Mk$. Trước tiên, ta tương ứng mỗi số nguyên dương s không vượt quá M với một phần tử của trường hữu hạn F_q. Việc đó dễ dàng làm được bằng cách sau đây. Giả sử $q=p^r$, và số s biểu diễn dưới cơ số p có dạng $s=(c_0,c_1,...,c_{r-1})_p$. Khi đó, đa thức

$$S(X)=\sum_{i=0}^{r-1} c_i X^i$$

modulo một đa thức bất khả quy nào đó bậc r sẽ tương ứng với một phần tử của trường F_q (xem Chương 5).

Như vậy, với m đã cho, với mỗi j, $1 \le j \le k$, ta có một phần tử tương ứng x_j của trường F_q. Ta sẽ chỉ ra một thuật toán để , với xác suất rất lớn, tìm được một x_j trong số đó sao cho tồn tại điểm (x_j,y_j) trên đường cong E. Khi đó, ta tương ứng số m với điểm $P_m=(x_j,y_j) \in E$ vừa tìm được.

Thuật toán tìm P_m:

El1. Đặt $j \leftarrow 1$.

El2. Nếu $j>k$: kết thúc thuật toán. Trong trường hợp ngược lại, đặt $Y_j \leftarrow x_j^3+ax_j+b$. Nếu tồn tại y_j sao cho $Y_j \equiv y_j^2 \pmod q$, in ra $P_m=(x_j,y_j)$ và kết thúc thuật toán. Nếu ngược lại, chuyển sang bước El3.

El3. Đặt $j \leftarrow j+1$ và quay về bước El2.

Vì mỗi phần tử $x \in F_q$, xác suất để $f(x)$ là chính phương bằng 1/2, nên thuật toán trên đây cho ta tìm ra điểm P_m với xác suất thất bại là $1/2^k$.

Như vậy, ta đã có một thuật toán để mã hoá m bằng cách tương ứng nó với một điểm của đường cong elliptic E. Tuy nhiên, cần nhắc lại rằng, một trong những yêu cầu của mã hoá là khi biết đường cong E trên F_q, biết P_m, ta phải khôi phục được m một cách dễ dàng. Trong trường hợp này, yêu cầu đó được đảm bảo. Thật vậy, giả sử $P_m=(x,y)$. Khi đó $m=\left[\frac{x-1}{k}\right]$ (trong đó [ ] là kí hiệu phần nguyên).

5.3. Mật mã khoá công khai sử dụng đường cong elliptic.

Trong chương 6, ta làm quen với một hệ mã khoá công khai, trong đó sử dụng độ phức tạp của phép tính tìm logarit cơ số b modulo p. Ở đây, ta có khái niệm hoàn toàn tương tự.

Giả sử B,P là các điểm của đường cong elliptic E, k là một số nguyên và $P=kB$. Khi đó ta nói k là logarit cơ sở B của P. Trong trường hợp E là đường cong trên trường F_q, $q=p^r$, $p\neq 2$, bài toán tìm logarit của các điểm trên một đường cong đòi hỏi thời gian mũ, và do đó, không thể thực hiện được trong khoảng thời gian chấp nhận được (nếu q được chọn đủ lớn).

Bây giờ giả sử có một tập hợp n cá thể cần trao đổi thông tin mật với nhau: $A_1,A_2,...,A_n$.

Trước tiên, ta chọn một đường cong elliptic E trên trường hữu hạn F_q với một điểm $B\in E$ dùng làm "cơ sở". Những thông tin này được thông báo công khai. Dĩ nhiên q phải là số đủ lớn.

Sau đó, mỗi cá thể A_j chọn cho mình khoá e_j, là một số nguyên nào đó. Khoá này được giữ bí mật, nhưng A_j thông báo công khai phần tử e_jB. Điều này không làm lộ khoá e_j do độ phức tạp của phép tính logarit.

Giả sử A_j cần gửi thông báo mật m cho A_i. Trước tiên, m được tương ứng với điểm $P_m\in E$ như đã trình bày ở trên. Sau đó, A_j sẽ chọn ngẫu nhiên một số s và chuyển cho A_i cặp điểm sau: $(sB, P_m+s(e_iB))$, nhờ e_iB đã được công khai. Khi nhận được cặp điểm này, A_i chỉ việc lấy số sau trừ đi e_i lần số trước để nhận được P_m:

$$P_m=P_m+ s(e_iB)- e_i(sB).$$

Chú ý rằng, chỉ có A_i làm được điều này vì e_i được giữ bí mật, và số s không thể tìm thấy trong thời gian chấp nhận được mặc dù biết sB, vì đó là logarit của (sB) cơ sở B.

Trong hệ mã vừa trình bày, ta không cần biết số N của đường cong E.

5.4. Hệ mã tương tự mã mũ.

Trong trường hợp này, các cá thể chọn chung cho mình một đường cong elliptic E trên trường hữu hạn F_q với N điểm. Các tham số này được thông báo công khai.

Để xây dựng hệ mã, mỗi cá thể A_i chọn cho mình khoá e_i, là số nguyên dương nằm giữa 1 và N, sao cho $(e_i,N)=1$. Bằng thuật toán Euclid, A_i tìm được d_i thoả mãn

$d_i e_i \equiv 1 (mod\ N)$. Bây giờ, giả sử A_i cần gửi thông báo m cho A_j. Cũng như trước đây, A_i tìm điểm P_m tương ứng trên đường cong. Sau đó,

1) Bước 1: A_i gửi cho A_j thông báo $e_i P_m$. Dĩ nhiên, khi nhận được thông báo này, A_j chưa thể giải mã vì không biết e_i và d_i.

2) Bước 2: A_j nhận thông báo được với e_j và gửi trả lại cho A_i thông báo $e_j(e_i P_m)$.

3) Bước 3: A_i lại gửi cho A_j thông báo sau khi đã nhân với d_i: $d_i\ e_j(e_i P_m)$.

4) Nhận được thông báo cuối cùng này, A_j nhân nó với khoá d_j của mình để nhận được $P_m = d_j d_i e_i e_j P_m$. Do cách chọn e_i, d_i, e_j, d_j ta có: $d_j d_i e_i e_j \equiv 1 (mod\ N)$, tức là $P = (1+sN)P_m$ với số nguyên s nào đó. Vì N là số điểm của đường cong nên $NP_m = 0$, và như vậy $P = P_m$. A_j đã nhận được thông báo ban đầu.

Để ý rằng, trong mỗi bước trên đây, các khoá mật e_i, d_i của các cá thể không hề bị phát hiện.

5.5. Chọn đường cong elliptic.

Có nhiều cách chọn đường cong và điểm B dùng làm "cơ sở" khi lập mã. Ở đây, ta trình bày hai cách đi theo hai hướng ngược nhau. Thứ nhất, chọn một điểm và một đường cong cụ thể. Thứ hai, lấy một đường cong trên trường số hữu tỷ và "sửa" theo modulo p khác nhau để thu được các đường cong trên trường hữu hạn.

Chọn đường cong và điểm ngẫu nhiên. Ta luôn luôn giả thiết rằng, đặc trưng của trường F_q khác 2, 3 (những trường hợp này có thể xét riêng). Khi đó, phương trình của đường cong có thể viết dưới dạng (7.2).

Giả sử x, y, a là ba phần tử lấy ngẫu nhiên của trường F_q. Ta đặt $b = y^2 - (x^3 + ax)$. Có thể kiểm tra dễ dàng đa thức $x^3 + ax + b$ có nghiệm bội hay không (xét biệt thức $4a^2 + 27b^3$). Nếu đa thức không có nghiệm bội, ta được đường cong E cho bởi phương trình

$$Y^2 = X^3 + aX + b$$

và điểm $B = (x,y) \in E$. Nếu đa thức có nghiệm bội, ta làm lại với một số a ngẫu nhiên khác.

Sửa theo modulo p. Ta xuất phát từ một đường cong elliptic E nào đó trên trường số hữu tỷ, và chọn $B \in E$ là một điểm bậc vô hạn. Sau đó, ta lấy một số nguyên tố p đủ lớn nào đó. Như đã nói, đường cong đã chọn chỉ có "sửa xấu"với một số hữu hạn số nguyên tố. Vì thế, nếu p chọn đủ lớn thì sửa theo modulo p sẽ cho ta đường cong elliptic E "modulo" p và điểm B modulo p. Cuối cùng, cũng chú ý là, cho đến nay, chưa có một thuật toán nào tương đối tốt để xác định số điểm N của một đường cong elliptic trên trường hữu hạn F_q với q là số rất lớn. Trong trường hợp N là tích cuả những số nguyên tố bé, có những thuật toán đặc biệt để tìm "logarit" cơ sở B, và do đó, hệ mã mà chúng ta đã xét sẽ không giữ được tính bảo mật nữa. Tuy nhiên, có nhiều phương pháp xác suất để tránh xảy ra tình trạng số điểm N của đường cong là tích của những số nguyên tố bé.

§6. L-hàm của đường cong elliptic

6.1. Như đã nói ở đầu chương, các đường cong elliptic có vai trò rất quan trọng trong nhiều vấn đề của số học. Tiết này có mục đích làm cho độc giả hình dung được phần nào ý nghĩa của đường cong elliptic trong Hình học đại số số học. Thực ra, đây là một lĩnh vực rất phong phú của toán học hiện đại. Vì thế, khó có thể trình bày trong một cuốn sách, hơn nữa, lại trong một giáo trình với yêu cầu khá sơ cấp. Chúng tôi cố gắng lựa chọn ở đây những kết quả và khái niệm cơ bản nhất, và cách trình bày là mô tả chứ không đi vào chi tiết.

Có thể nói, khái niệm quan trọng nhất trong nghiên cứu đường cong elliptic là *L-hàm*. Giả sử ta xét đường cong elliptic trên trường số hữu tỷ Q. Nếu cần thiết thì khử mẫu số ở các hệ số của phương trình xác định đường cong, ta cố thể giả thiết ngay tư đầu rằng, đường cong được cho bởi phương trình với các hệ số nguyên.

Để nghiên cứu đường cong cho trên trường số hữu tỷ, người ta *nghiên cứu đồng thời các sửa theo modulo p* của đường cong đó ứng với mọi số nguyên tố p. Ta nhắc lại rằng, đó là các đường cong nhận được bằng cách thay các hệ số bởi các thặng dư modulo p của chúng. Có thể tồn tại một số hữu hạn số nguyên tố p tại đó đường cong nhận được có điểm bội. Trước hết, ta xét các số nguyên tố p tại đó đường cong có " sửa tốt", tức là ta có đường cong elliptic trên trường F_p.

6.2. L-hàm của đường cong elliptic trên trường hữu hạn.

Giả sử E là đường cong elliptic trên trường số hữu tỷ Q, có sửa tốt tại số nguyên tố p. Đồng thời với việc xét E modulo p, ta có thể xét các điểm của đường cong E trên mọi trường F_q, với $q=p^r$, $r=1,2,...$Kí hiệu qua N_r số điểm của đường cong E trên F_q, $q=p^r$.

Như vậy, ta có dãy các số nguyên dương $N_1,N_2,...N_r...$ Để nghiên cứu một dãy nào đó, một trong những phương pháp rất hay được dùng trong số học là xét *hàm sinh* của chúng. Nhờ hàm sinh, người ta có thể xét các phần tử của dãy một cách đồng thời, thông qua tính chất của hàm sinh. Chẳng hạn, hàm sinh của dãy số trên đây là:

$$Z_p(T)=\exp\left(\sum_{r\geq 1}\frac{N_r}{r}T^r\right) \qquad (7.10)$$

Định nghĩa. Hàm $Z_p(T)$ được gọi là *Zeta-hàm* của đường cong E trên trường F_p.

Zeta-hàm được xây dựng không chỉ với các đường cong elliptic, mà còn với những đối tượng rộng hơn, là các đa tạp xạ ảnh. Zeta-hàm của các đa tạp xạ ảnh có nhiều tính chất tương tự với Zeta-hàm Riemann. Một trong những tính chất quan trọng nhất của các Zeta-hàm thể hiện trong định lí sau đây, mà ta chỉ phát biểu cho các đường cong elliptic.

Định lí Weil. *Zeta-hàm của một đường cong elliptic E trên trường hữu hạn F_q là một hàm hữu tỷ của T, có dạng:*

$$Z(T;E/F_q)=\frac{1-aT-qT^2}{(1-T)(1-qT)},$$

trong đó a là số tham gia trong công thức tính số điểm của đường cong E trên F_p: $N_1=1+q-a$. Định thức của đa thức ở tử số âm, và hai nghiệm (phức liên hợp) của nó có trị tuyệt đối bằng $\sqrt{q}$.

Nhận xét. 1) Khi biết Zeta-hàm, ta có thể khai triển để tìm các hệ số của nó trong công thức (7.10), nghĩa là biết được số điểm của E trên trường $F\,p^r$ với mọi r tuỳ ý. Vì Zeta-hàm chỉ phụ thuộc $a=1+q-N_1$ nên N_r xác định duy nhất qua N_1.

2) Tính chất định thức của tử số âm có nghĩa là: $|a|<2\sqrt{q}$. Như vậy, định lí Hasse là một hệ quả của định lí Weil.

Một tương tự của định lí Weil cho các đa tạp xạ ảnh được gọi là "giả thuyết Weil", và được P. Deligne chứng minh năm 1973.

6.3. L-hàm của đường cong trên trường số hữu tỷ

Như vậy, với mỗi số nguyên tố p, ta có Zeta-hàm $Z_p(T)$ ứng với đường cong elliptic E trên trường F_q, $q=p^r$. Để nghiên cứu đường cong E trên trường số hữu tỷ, ta có thể xét "đồng thời" các "Zeta-hàm địa phương" $Z_p(T)$ bằng cách xây dựng "Zeta-hàm toàn cục" của biến phức s.

Kí hiệu qua a_p số xác định bởi $a_p=p+1-N_p$, trong đó N_p là số điểm của đường cong trên trường F_p. Khi đó L-hàm Hasse-Weil của đường cong E được định nghĩa bởi công thức

$$\mathrm{L(E,s)}=\prod_p (1-a_p p^{-s}+p^{1-2s})^{-1}\ .$$

Nhận xét. Bằng cách khai triển tích trong định nghĩa L-hàm, ta có thể thấy rằng, L-hàm tương tự như Zeta-hàm Riemann, định nghĩa bởi công thức sau đây:

$$\zeta(s)=\sum_{n=1}^{\infty} n^{-s}\ .$$

Dễ thấy rằng, Zeta-hàm Riemann có thể được tính bởi công thức sau (xem phần bài tập):

$$\zeta(s)=\prod_{p\in P}(1-p^{-s})^{-1}$$

Đối với Zeta-hàm Riemann, các tính chất quan trọng nhất là:

1) $\zeta(s)$ có thể thác triển thành hàm phân hình trên toàn mặt phẳng phức với cực điểm đơn tại $s=1$.

2) Nếu đặt

$$\Lambda(s) = \pi^{-s/2}\Gamma(s/2)\zeta(s)$$

thì $\Lambda(s)$ là hàm phân hình trên toàn mặt phẳng phức, và thoả mãn phương trình hàm

$$\Lambda(s) = \Lambda(1-s)$$

3) $\zeta(-2n) = 0$ với mọi n nguyên dương.

Giả thuyết Riemann nổi tiếng nói rằng, các không điểm còn lại của Zeta-hàm đều nằm trên đường thẳng *Re s=1/2*. Người ta đã kiểm tra giả thuyết đó đối với một số rất lớn không điểm (hàng triệu), nhưng vẫn chưa chứng minh được giả thuyết trong trường hợp tổng quát. Giả thuyết này cũng liên quan đến nhiều vấn đề của số học thuật toán.

Đối với L-hàm của đường cong elliptic, ta có:

Giả thuyết: Hàm *L(E,s)* có thể thác triển giải tích lên toàn mặt phẳng phức. Hơn nữa, tồn tại một số nguyên dương N sao cho, nếu đặt

$$\Lambda(E,s) = N^{s/2}(2\pi)^{-s}\Gamma(s)L(E,s),$$

thì ta có phương trình hàm sau đây

$$\Lambda(E,2-s) = \pm\Lambda(E,s).$$

Số N nói trong giả thuyết là một bất biến quan trọng của đường cong, gọi là *conđuctơ* của nó.

6.4. Giả thuyết Birch-Swinnerton-Dyer.

Một trong những giả thuyết quan trọng khác của lí thuyết các đường cong elliptic là giả thuyết sau đây của Birch và Swinnerton-Dyer.

Trước tiên ta nhắc lại rằng, nhóm các điểm hữu tỷ của đường cong elliptic E là tổng trực tiếp của nhóm cấp hữu hạn với nhóm Z^r. Số r được gọi là hạng của đường cong.

Giả thuyết Birch-Swinnerton-Dyer. Nếu hạng của đường cong elliptic E bằng r thì L-hàm của đường cong có không điểm cấp r tại điểm $s=1$.

Như vậy, nếu $L(E,1)=0$ thì hạng $r \geq 1$, và do đó, đường cong elliptic có vô hạn điểm hữu tỷ. Trong trường hợp ngược lại, đường cong E chỉ có hữu hạn điểm hữu tỷ. Đó chính là kết quả quan trọng đầu tiên theo hướng khẳng định giả thuyết Birch-Swinnerton-Dyer, được Coates và Wiles chứng minh năm 1977.

Thực ra, giả thuyết Birch-Swinnerton-Dyer còn cho công thức tính giới hạn

$$\lim_{s\to 1}(s-1)^r L(E,s).$$

Tuy nhiên, để phát biểu chính xác giả thuyết đó ta cần đến khái niệm nhóm Shafarevich của đường cong, là một trong những khái niệm sâu sắc nhất của hình học đại số.

Nhận xét. Để kết thúc chương này, chúng tôi xin nói qua vài lời về giả thuyết quan trọng nhất trong lí thuyết đường cong elliptic: giả thuyết Taniyama-Weil.

Giả sử N là một số nguyên dương. Ta kí hiệu qua nhóm $\Gamma_0(N)$ nhóm các ma trận vuông cấp 2 $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ trong đó a,b,c,d nguyên, $ad-bc=1$ và $c \equiv 0 (mod\ N)$. Nhóm các ma trận này tác động lên nửa mặt phẳng trên theo công thức sau:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}(z)=\frac{az+b}{cz+d}.$$

Một hàm $f(z)$ giải tích tại nửa mặt phẳng trên, kể cả tại các điểm hữu tỷ của trục thực và bằng không tại các điểm đó, được gọi là một *dạng modula trọng số 2 đối với nhóm* $\Gamma_0(N)$ nếu nó thoả mãn hệ thức sau:

$$f(\gamma z)=(cz+d)^2 f(z).$$

Từ định nghĩa trên suy ra rằng, nếu $f(z)$ là dạng modula trọng số 2 đối với nhóm $\Gamma_0(N)$ thì ta có $f(z+1)=f(z)$ với mọi z (lấy γ là ma trận $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$. Như vậy, $f(z)$ có thể khai triển theo dạng sau:

$$f(z) = \sum_{n=1}^{\infty} a_n e^{2\pi i n z}.$$

Từ đó ta có thể tương ứng f với L-hàm của nó:

$$L_f(s) = \sum_{n=1}^{\infty} a_n / n^s$$

Giả thuyết Taniyama-Weil: Nếu E là một đường cong elliptic trên trường số hữu tỷ côndưctơ N thì L-hàm của đường cong E là L-hàm của một dạng modula trọng số 2 đối với nhóm $\Gamma_0(N)$.

Giả thuyết trên đây liên quan chặt chẽ đến định lí lớn Fermat. Thật vậy, giả sử tồn tại số nguyên tố p lớn hơn 2 sao cho phương trình Fermat với số mũ p có các nghiệm không tầm thường a, b, c.

Ta đổi dấu c và viết phương trình dưới dạng:

$$a^p+b^p+c^p=0$$

Đường cong elliptic xác định bởi phương trình

$$y^2=x(x-a^p)(x+b^p)$$

có côndưctơ $N=N_0(abc)$ (xem định nghĩa N_0 ở chương 5). Đường cong này được G. Frey nghiên cứu lần đầu tiên năm 1983. Sau đó (1986), K. Ribet chứng minh rằng, L-hàm của đường cong đó không phải là L-hàm của bất kì một dạng modula trọng số 2 nào đối với nhóm $\Gamma_0(N)$. Như vậy, nếu chứng minh được giả thuyết

Taniyama-Weil thì cũng chứng minh được định lí lớn Fermat, bởi vì nếu phương trình Fermat có nghiệm thì tồn tại một đường cong elliptic không thoả mãn giả thuyết Taniyama-Weil.

Tháng 6 năm 1993, A.Wiles công bố chứng minh giả thuyết Taniyama-Weil, cũng tức là chứng minh được định lí lớn Fermat.

Bài tập chương 7

7.1. Cho dường cong elliptic trên trường thực $y^2=x^3-36x$ và các điểm trên đường cong: $P=(-3,9)$, $Q=(-2,6)$. Hãy tính các điểm $P+Q$ và $2P$.

7.2. Tìm bậc của điểm $P=(2,3)$ trên đường cong $y^2=x^3+1$.

7.3. Chứng minh rằng các đường cong elliptic sau đây có $q+1$ điểm trên trường F_q:

1) $y^2=x^3-x$, $q \equiv 3 (mod\ 4)$.

2) $y^2=x^3-1$, $q \equiv 2 (mod\ 3)$, q lẻ.

3) $y^2+y=x^3$, $q \equiv 2 (mod\ 3)$, (q có thể chẵn).

7.4. Tính số điểm của đường cong elliptic $y^2=x^3-x$ trên trường F_{71}.

7.5. Cho đường cong $y^2+y=x^3$ trên trường F_2. Hãy tìm Zeta-hàm của đường cong đó và tính số điểm của đường cong trên mọi trường F_q với $q=2^r$, $r=1,2,...$