解　説

数論アルゴリズムとその応用

数論アルゴリズムの研究動向†

一　松　　信††

1. 序

本特集は，符号理論や公開鍵暗号などの応用に際して重要な役割りを果たす数論（整数論）関係のアルゴリズムの紹介を目的とする．その具体例として，素数判定アルゴリズム，素因数分解及び離散対数アルゴリズム，有限体上の諸演算（累乗根，代数方程式の解など），多項式の因数分解，代数幾何学的アルゴリズム，ならびにこれらの基礎として多倍長演算や剰余演算などの効率的な諸算法がある．それらの詳細は，各論文にゆずることとし，入門的な準備と解説を **2.**, **3.** で論じる．もちろん符号理論や公開鍵暗号などへの応用も，本特集の重要な話題である．

コンピュータの発展により「やってみるしかなかった」組合せ問題などの多くが解決された．しかし有限の対象なので，解の存在は自明だが，効率的な解法がなく，いわば「時間が勝負」といった一群の問題が論じられてきた．パラメタ n に対して n の多項式的な時間で解ける「効率的」な解法のある**P問題**や，分岐が正しく選べれば n の多項式的な深さで解ける（あるいは検算ができる）非決定性多項式時間問題——略して**NP問題**——などは，計算量の理論でおなじみであろう．

数論アルゴリズム中には，完全な調査は面倒だが「うまくいって答がでれば儲けもの」といった方式の，実用上では有用な「確率的算法」が多数知られている．これらは厳密な意味では「算法（アルゴリズム）」と呼ぶのに抵抗を感じるが，現在ではその種の手法も「拡張された算法概念」に含めて考えるのが普通である．それらを抽象化した計算量の諸クラスもあり，**5.**, **6.** で紹介する[9),18)]．

このような研究は，一面では計算機の高速化・小型化に支えられて発展したが，他方計算機の設計方針にもフィードバックされている．円周率を20億桁計算するのは特殊な例であるが，桁数ならびに指数部の範囲を事実上無制限にした「多倍長計算」の必要性・重要性が，ようやく数学者以外の人々にも認識され始めたのを喜ばしく思うのは，筆者の偏見だろうか．

余談ながら秘密通信・データの保護などの技術は，一種の総合科学であって，利用できるものは何でも使おうという貪欲な分野である．「役にたたない理論」の代表とされた整数論が応用されて，本特集になったのがその一例であるが，近年量子力学の不確定性原理を活用した「量子暗号」の提案がある[10)]．それは途中で盗聴（盗視）されれば光子の状態に変化が生じて，それが検出できることを逆用して，通信路の安全を確認しようという着想である．現在のところまだ理論的興味にすぎないが，既成の分野にとらわれず，何が現れても驚かない柔軟な心構えが必要なのだろう．

2. 整数論の基礎

以下の記述は，整数論の教科書（たとえば 1)〜5)）にある内容の要約であり，多くの読者は既知であろう．

二つの整数 a, b があり，$a=b\cdot c$ である整数 c が存在するとき，a を b の**倍数**，b を a の**約数**または**因数**といい，このとき a は b で**割り切れる**，または b が a を**整除する**といって $b|a$ で表す．二つの整数 a, b の共通な約数を**公約数**といい，そのうち正で最大なもの d を**最大公約数**といって，GCD (a,b) で表す☆．$n|(a-b)$ のとき，$a\equiv b$

† Recent Tendency of Study on Number-Theoretic Algorithms by Sin HITOTUMATU (Tokyo Denki Univ., HATOYAMA Campus).

†† 東京電機大学理工学部

☆ 教科書には伝統的に GCM (greatest common measure) と書いてある場合が多いが，現在の英語では GCD (greatest common divisor) が慣用である．

$(\mathrm{mod}\, n)$ と記し，a と b とは n を法として**合同**という．

乗法の単位元である1の約数 ± 1 を**単数**という．単数でない整数 p で，単数と $\pm p$ 以外に約数をもたないものを**素数**といい，単数でも素数でもない整数を合成数という．通常は正の整数だけを考えるが，素因数分解の算法では負の整数も扱うと便利なことが多い．そのときには $p_0=-1$ を一つの特別な素数とみなして処理することが多い．

合成数 n は素因数に分解される．同一の素数の項をまとめると，$p_0^{s}p_1^{e_1}\cdots p_r^{e_r}$ ($p_0=-1$, s は0か1) の形になる．この分解は，p_i の順序を問わなければ本質的に一通りである（**初等整数論の基本定理**）．

任意の二つの整数 a, b に対して，$b\neq 0$ ならば

$$a=q\cdot b+r,\quad 0\leqq r<b$$

である q, r がただ一つずつ定まる．q を，a を b で割った**商**または**整商**，r を**余り**または**剰余**という．通常は余り r を $0\leqq r<b$ と標準化するが，負の余りも許して $-b/2\leqq r<b/2$ ととることもある．この場合，**絶対値最小の剰余**という．例 $23\div 8=3$ 余り (-1)．互除法の演算においてこのように修正すると，高速化ができる．

互除法の算法は次のとおりである．

1° 与えられた整数 m, n に対し，$n_0:=m$; $n_1:=n$ とおく．

2° 除法 $n_{i-1}=q_i\cdot n_i+n_{i+1}$, $0\leqq n_{i+1}<n_i$ を実行する $(i=1,2,\cdots)$．

3° $n_{i+1}=0$ ならば $d:=n_i$ で完了（必ず有限回で完了）；そうでなければ次の i について2°へ戻る．

実際には $a:=m$; $b:=n$; $a\div b=q$ 余り r とし，$r\neq 0$ ならば $a:=b$; $b:=r$ として次へ進めばよい．整商 q は不要だが，もしも

$$u_0:=1,\quad u_1:=1,\quad v_0:=0,\quad v_1:=1$$

として毎回除法のつど

$$u_{i+1}:=u_{i-1}-u_iq_i;\quad v_{i+1}:=v_{i-1}-v_iq_i$$

を実行すれば，最後に $n_{t+1}=0$, $n_t=d$ となったとき，$u_tm+v_tn=d$ である．これにより m, n が互いに素なとき，$um\equiv 1\ (\mathrm{mod}\, n)$ である u (n を法とする m の**逆数**）が計算できる．

正の整数 p を定めると，それを法とした剰余系 $\mathbf{Z}_p=\{0,1,\cdots,p-1\}$ には自然に加法・乗法が定義されて環になる．p が素数ならば，0以外の要素には逆数があるので体になる（**標数 p の有限素体**）．任意の有限体 K は，標数 p の素体 $\mathbf{Z}_p$ の有限次拡大体であり，要素の個数 p^l によって構造が一意的に定まる．符号理論では $p=2$ の場合が多い．

複数の素数 $p_1,\cdots,p_r$ に対して，$a_i\in\mathbf{Z}_{p_i}$ $(i=1,\cdots,r)$ を任意に与えると，各 i について $a\equiv a_i\ (\mathrm{mod}\, p_i)$ である a: $0\leqq a<p_1\cdots p_r$ が一意的に定まり，帰納的に計算できる（**孫子剰余定理**）*．

p と互いに素な a に対して，$x^k\equiv a\ (\mathrm{mod}\, p)$ であるような x が存在するとき，a を p に対する ***k* 乗剰余**といい，x が存在しないとき ***k* 乗非剰余**という．$k=2$ のときは**平方**とよぶ．a が p の平方剰余か否かを，$(a|p)=+1$ または -1 で表す（**Legendre の記号**）．p が奇素数のとき，$(a|p)\equiv a^{(p-1)/2}\ (\mathrm{mod}\, p)$ である．$(a|p)$ は次の諸公式で容易に計算できる．

$(ab|p)=(a|p)(b|p)$,

$a\equiv b\ (\mathrm{mod}\, p)$ なら $(a|p)=(b|p)$

p, q が奇素数なら $(p|q)(q|p)$

$=(-1)^{(p-1)(q-1)/4}$ （平方剰余の**相互法則**）

$(-1|p)=(-1)^{(p-1)/2}$, $(2|p)=(-1)^{(p^2-1)/8}$

（第1及び第2**補充法則**）

3. 素因数分解算法

素数と素因数分解は，小学校以来おなじみであろうが，その具体的算法については，きわめて小さい整数に使われる素朴な算法——2, 3, 5, …で次次に割れるだけ割る——以上の手法を習わなかった人々が多いのではあるまいか．このような素朴な方法では，いくら計算機が高速度になっても，すぐに壁にぶつかる．

互除法による二つの整数 m, n の最大公約数の計算は，最悪でも m, n の小さい数の十進桁数の5倍以内で完了する（**Lamé の定理**）[3]，もし毎回剰余に負も許して絶対値最小の余りにすれば，その 60% 以下で済む．このように互除法はきわめて効率的な算法なので，大きな整数 p を素因数分解するには，p と公約数をもつ（可能性が高い）数 q を求めて，p, q の最大公約数を互除法で計算するのが基本になる．

そのためによく使われるのが，楕円曲線法と2

---

* 英語では Chinese remainder theorem（ある中国人の剰余定理）というが，中国での慣例に従う．

次ふるい法である．詳細は「素因数分解と離散対数問題アルゴリズム」にゆずるが，後者に対して一言する（たとえば文献 2)，4)）．これは整数 p に対して

$$m^2\equiv n^2 \pmod p,\quad m\not\equiv \pm n \pmod p$$

である m，n がみつかれば，p は素数でなく，p と $m+n$ または $m-n$ の最大公約数を計算して，p の因数が分かる．これが**2次ふるい法**の原理である．近年 $b^2-4ac=kp$（k は小さい整数）を満たす多数の2次式 ax^2+bx+c から，上記の性質を満たす m，n の組を組織的に求める方法が，**複数多項式2次ふるい法**として活用されている．この算法は，いわば多数の情報を蒐集する特派員（並列処理可能）と，その情報を整理して役に立つ m，n の組を計算する編集部との協力作業である．いかに多数のデータを集めても，起源が同一のネタの変形ばかりではなんの役にも立たない．

ところで有限体の0以外の要素は，乗法に関して巡回群をなす．その生成元を**原始根**という．原始根の存在証明や理論上の構成法は難しくないが[1]，そのままでは効率的でない．

素数 p に対する原始根 r を一つ定めたとき，$a\equiv r^b \pmod p$ ならば，$b=\log_r a$ と記して，a の r に対する**離散対数**という．実際通例の対数と似た性質がある．昔は**指数**とよんで $\mathrm{Ind}_r a$ と記したが[1]，現在では「対数」のほうが慣用である．

p，r を定めたとき，b から a を計算するのは，たとえば b を二進展開して2乗を反復すれば $2\log_2 b$ 以下の手間で容易にできるが，a を与えて b を計算するのは一般に難しい．この性質が公開鍵暗号，あるいは鍵の交換プロトコルに活用されている[15]．通信者 A，B が，p，r を共通にもった後，それぞれ乱数 x，u を用意し，A は B に $y=r^x \pmod p$ を，B は A に $v=r^u \pmod p$ を送れば，A は v^x，B は y^u を計算して共通の値 $r^{xu}=v^x=y^u$ をもつことができるが，第三者が（A，B自身も）秘密の乱数 x，u を知ることは困難である．

うまくいけば幸いという確率的な素因数分解算法として，Pollard の **ρ 法**がある[20]．整数 n に対して

数列　$x_0=1$，$x_{k+1}\equiv x_k^2+1 \pmod n$

（x_k^2-1，x_k^3+1 などの変形も使われる）

を作れば，そのうちに n の約数 p に対して

$$x_k\equiv x_l \pmod p \quad (k<l)$$

になることがあり，以後 $l-k$ が周期になる．$l-k$ の倍数 m を十分大にとれば，$x_m\equiv x_{2m} \pmod p$ となり，$\gcd(x_{2m}-x_m, n)$ によって，n の約数が分かる可能性が高い．m は不明だが，十分大きな k に対し，$(x_{2k}-x_k)$ と相続く数十個の k について積をとり，それと n との最大公約数を計算するという使い方をする．「ρ 法」という名は，x_1 から始めた次々の x_k の列が，ある所で以前の値と合流して以後周期的になるのが，丸にしっぽをつけた ρ の象型だからである．

現在の段階で，70桁程度の整数の素因数分解は，パソコンでも可能であり，特殊な数では150桁の整数の素因数分解も成功している[4),6)]．しかし特別な性質のない一般の整数に対しては，100桁程度が限界らしい．もちろん計算機の高速化により，この限界は急激に上昇しつつある．

4. 素数判定法

素因数分解ができれば，整数 n はもちろん素数ではないが，n が素数かどうか——合成数と分かっても素因数は分からなくてよい——という判定だけならば，直接のずっと早い算法が多い．

p が素数なら，p が割り切れない任意の整数 a に対して，$a^{p-1}\equiv 1 \pmod p$ である（**Fermat の小定理**）[1)~5)]．この性質はいろいろな方法で証明できる．たとえば $(a+b)^p\equiv a^p+b^p \pmod p$ により，$a^p\equiv a \pmod p$ を a に関する数学的帰納法で示すことができる．したがってもし

$$p\nmid a \text{ でなく，} a^{p-1}\not\equiv 1 \pmod p$$

である a があれば，p は素数でない（**Fermat テスト**）．これは簡便な予備テストとして広く使われるが，素数でないと判定されても素因数は分からないことと，逆は正しくないことに注意する．

ただし $p-1$ が完全に素因数分解され，その各素因数 q_i に対して

$$a_i^{(p-1)/q_i}\not\equiv 1,\quad a_i^{p-1}\equiv 1 \pmod p$$

である a_i が存在すれば，**p は素数**である．$p-1$ が完全に素因数分解できなくても，q_i を素数として

$$p-1=q_1^{e_1}\cdots q_r^{e_r}\cdot m,\quad m<\sqrt{p},$$

各 i に対し $a_i^{p-1}\equiv 1 \pmod p$，かつ

$$\gcd(a_i^{(p-1)/q_i}-1, p)=1$$

である a_i が存在すれば（m が素数であるかどう

かは不問のまま)，**pは素数**である．上記のa_iは，多くの場合 2, 3, 5, … などの小さい素数や，いくつかの乱数を試みて成功する（たとえば文献 2), 4)).

$p+1$ の素因数がすべて既知の場合にも，効率的な素数判定法がある．Mersenne 数 2^n-1 に対する Lucas テスト（後述）がその典型例である．1992 年の初め，32 番目の Mersenne 素数 $2^{756839}-1$ (227832 桁) の発見が報道された．現在のところ最大の双子素数とされる $1706595\times 2^{111235}\pm 1$ も，この種の方法で求められたものである．なお平方剰余のテストも有用である[13]．

一般の **Lucas テスト**を述べる．p が大きな奇数で，$p+1=q_1^{e_1}\cdots q_r^{e_r}$ と素因数分解できたとする．

$\gcd(p, b(a^2+4b))=1$,

a^2+4b が p を法として平方非剰余

である整数 a, b を選び，**一般 Fibonacci 数列**

$y_0=0$, $y_1=1$, $y_{k+1}=ay_k+by_{k-1}$ $(k\geqq 2)$

を作るとき，$i=1,\cdots,r$ のすべてについて

$y_{p+1}\equiv 0 \pmod p$, $y_{p+1}/q_i\not\equiv 0 \pmod p$

ならば，p は素数である．y_k は漸化式

$y_{2k}=2y_ky_{k+1}-ay_k^2$

$y_{2k+1}=y_{k+1}^2+by_k^2$

$y_{2k+2}=ay_{k+1}^2+2by_ky_{k+1}$

により，$\log k$ に比例した手間で計算できる．

以上は特別な数 p についてだが，任意の奇数 p に対しても，素数の累乗でなければ（これは容易に検査できる），ある限界 B まで，素数の列 $a=2, 3, 5, 7, \cdots$ に対して次の検査を行い，すべて正しければ，p は素数である (Miller の判定法)[19]．

(i) $a\nmid p$ でない　(ii) $a^{p-1}\equiv 1 \pmod p$

(iii) $r=(p-1)/2^k$ が奇数になるまでの各 $k=1, 2, \cdots$ に対し，$\gcd(a^r-1, p)=1$.

限界 B は p が素数でないとしたとき，その素因数に関する最小の平方非剰余であり，理論上 $p^{0.134}$ で評価される．もし ERH（拡張された Riemann 予想）が正しければ $\lceil \mathrm{lb}\, p\rceil^4 \cdot \mathrm{lb}\ \mathrm{lb}\ \mathrm{lb}\, p$ ($\mathrm{lb}=\log_2$) という，p の桁数の多項式の手間で済む．

5. 素数判定の計算量

整数 n に関する算法の計算量は，通例 $x=\log n$ (n の桁数) について考える．したがって互除法は P 的算法だが，素数判定や素因数分解を $\sqrt{n}$ までの素数で次々に割って検査する素朴な算法の手間は $e^{x/2}$ であって，P 的算法ではない．

素数判定は NP 困難か？—そうではないらしい．もしそうなら Miller の算法は否定され，EHR は正しくないことになる．さらに素数判定は

NP∩co-NP (Pratt の定理)[21]

なので，そうだとすれば NP=co-NP という信じ難い結果をうる．

確率的算法の計算量として Johnson の **RP** (Randomly Polynomial)[18] がある．それは乱数を使って多項式時間で動く算法Aがあり，もしも答が正（たとえば p は素数）のとき，Aによって正という解答がでる確率が 1/2 以上であり，逆に答が否（たとえば p が合成数）のときには，確実に否と判定される場合である．Solovay-Strassen の算法[26]やその修正は RP 算法である．

逆の方向の判定には，ほとんどすべての素数に有効な，楕円曲線を利用した Goldwasser-Kilian の算法及びその改良があり，1987 年に Adleman-Huang が，素数判定は RP∩co-RP=ZPP というクラスに属することを証明した[8]．これはPともNPとも異なる両者の中間の族と予想されている．ただしその算法の手間は，定数係数も多項式の次数もきわめて大きく，実用的とはいい難い．

RP の定義で，正のとき正と解答する確率を 3/4 以上とし，否のとき確実に否と判定できないが誤まる確率が 1/4 以下，と修正したクラスを BPP という（定数 3/4, 1/4 は若干修正してよい；Gill)[16]．Miller-Rabin のテスト[22]は BPP 算法の一例である．若干の自然な仮定の下で，RP あるいは BPP に属する対象は，決定論的な準指数的な手間で解けることが証明されている (Yao[27], Boppana-Hirschfied[12])．70 桁の整数 123456789 123456789 123456789 123456789 123456789 123456789 123456789 1234567 が素数であることは，現在のパソコンで 1 分強で検証できるが，スーパコンピュータでは，300 桁の整数の素数判定が数分で実行可能である．

6. 素因数分解と離散対数の計算量

大きな整数 n の素因数分解の手間が，素数性判定よりもずっと多いという経験的事実が，公開鍵暗号，特に RSA 系暗号の基礎である[7]．しかしそ

の差は理論的に証明されてはいない.

素因数分解はNP問題であるが,またco-NP問題でもある.したがってもしもNP-完全だとすると,NP=co-NPという信じ難い結論になるので,たぶんそうではないらしい.

これまでに得られた最も速い決定的な算法は,$n^{1/4+O(1)}$ のオーダであるShanksの算法[25]である.もしEHRが正しければ,この指数が $1/5+O(1)$ に減らせる(Schoof)[23].他方確率的算法では,

$$L(n)=\exp\sqrt{\log n\cdot\log\log n}$$

あるいはそれを1に近い数乗した時間で可能な方法がいくつか知られている.ρ 法も n の因数 p を $(\log^2 p)/p$ の定数倍以上の確率で経験的に $n^{1/4+O(1)}$ 以下の時間で発見する.$L(n)$ は確かに準指数的だが,$n=10^{100}$ 程度では,$n^{1/5}$ と大差がない.ただ $L(n)$ が手間の一般的な下限らしいという予想がある(数体ふるい法の発見により,この予想は,現在では $L(n)\to L'(n)=\exp\sqrt[3]{\log n\cdot\log\log n}$ に修正されている).もしもこれが正しければ,P≠NPである.素因数分解の計算量は,PとNP-完全との中間にある新しい興味あるクラスらしい.

離散対数も,素数 p に対してほぼ $L(p)$ 程度の手間で計算する算法が知られているが,素因数分解と同じクラスなのか,また両者の手間になんらかの関係があるのかどうかは未知である.

十分の記憶があるときに有用なのは「対数表」(データベース)を作る方法である(たとえば文献14)).原始根 r の乱数 b 乗 $r^b\equiv a\pmod p$ を作り,a を素因数分解する.このような値が十分多数えられれば,連立一次方程式を解いて,小さい素数に対する離散対数を計算することができる.与えられた a に r の乱数乗を掛け,もしその結果が表にある素数だけの積で表されれば,それから $\log_r a$ が計算できる.類似の方法は群の生成元による群の要素の表現にも活用できる.

この種の算法が,離散対数を活用したEl Gamal系の暗号[15]をグループで共有する鍵によって使う閾値暗号において,グループの構成員が他の仲間の秘密鍵を知りうる手掛りになるか(第三者よりも少ない手間で発見できるか)は不明である.少なくともこの種の暗号を使用するときには,グループ内の良心的相互協力を暗黙の前提に仮定する必要があるように思う.

7. 関連した話題

次のような課題も,数論的算法のうちに含ませて考えることが多い.

有限体の具体的構成

有限体上の代数方程式の解(平方根など)

有限体上の多項式の因数分解

特に多項式の因数分解は,通常の整数係数多項式を因数分解するための補助手段としても重要であり,Berlekampの算法[11]が有名である.ただしこれは(実用上では効率的であるが),一般的には有限体の要素数に比例する手間がかかり,完全なP的算法ではない.

その意味でSchoofの算法[24]は,固定した多項式に対し,素数 p を法とした2次式に因数分解する多項式時間の決定的算法という点で注目に値する.多項式時間の次数が高く,必ずしも実用的ではないといわれるが,これにより永らく懸案とされてきた,$4m+1$ 型の素数 p を a^2+b^2 と(一意的に)分解する計算が,P問題であることが確かめられた.—これは p を法とした -1 の平方根 r を知れば,p と $r+i$ とにGaussの整数の互除法を施して,容易に計算できる(たとえば文献2),3)).$6m+1$ 型の素数 p を a^2+ab+b^2 と $(a,b>0$ とすれば一意的)分解する計算も同様である.

8. むすび

以上ごく表面を眺めただけで,本特集の序論にすぎないが,最後に数学教育との関連について一言したい.

近年日本では中等段階での代数学が軽視される傾向にある.これはあまりにも繁雑な計算が入試に氾濫した反動だろうが,計算機による数式処理の進展とあいまって,多項式代数の重要性は,増加しこそすれ,けっして減少していない.文字式の代数は,いまや算数につぐ「大衆の数学」とされる.本特集はかなり特殊な話題だが,純粋数学者が計算機に親しみをもち,代数学を見直してくださる一つの契機になることを期待する.

参考文献

全般的な教科書(及び数表)

1) 高木貞治:初等整数論講義,共立出版,初版1931,改訂版1967.

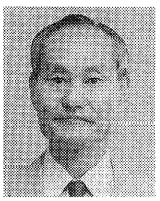
2) 和田秀男:コンピュータと素因子分解,遊星社(1987).

3) 一松 信：代数学入門第一課，近代科学社 (1992).
4) 森本光生・木田祐司他：円分数の素因数分解，上智大学数学講究録，Ⅰ，1987，Ⅱ，1989，Ⅲ，1992 (一部英文).
5) Riesel, H.: Prime Numbers and Computer Methods for Factorization, Birkhauser (1985).
6) Brillhard, J. et al.: Factorizations of b^r+1, $b=2, 3, 5, 6, 7, 10, 11, 12$ up to high powers, Amer. Math, Soc. Providence 1983, 改訂版 1988.
7) Salomaa, A.: Public-Key Cryptography, Springer-Verlag, 1990, 日本語訳，足立暁生訳，東京電機大学出版局 (1992).

論文

8) Adleman, L. M.-Huang, M. D.: Recognizing Primes in Random Polynomial Time, Proc. 19th Ann. ACM Symp. Theory Comp., pp. 462-469 (1987).
9) Bach, E.: Number-Theoretic Algorthms, Ann. Rev. Computer Sci. pp. 119-172 (1990. 4).
10) Bennett, C.H-Brassard, G.-Ekert, A.K.: Quantum Cryptography, Scientific American. pp. 50-59 (Oct. 1992).
11) Berlekamp, E. R.: Factoring Polynomials over Finite Field, Bell. Sys. Tech, J. 46, pp. 1853-1859 (1967).
12) Boppana, R. B.-Hirschfield, R.: Pseudorandom Generators and Complexity Classes, Advances in Computing Research, 5 (1985).
13) Cohen, H.-Lenstra, H. W. Jr.: Primality Testing and Jacobi Sums, Math. Comp., 42, pp. 297-330 (1983).
14) Coppersmith, D.-Odlyzko, A. M.-Schroeppel, R.: Discrete Logarithms in GF(p), Algorithmica 1, pp. 1-15 (1986).
15) ElGamal, T.: On Computing Logarithms over Finite Field, Proc. CRYPTO'85, Springer-Verlag, pp. 396-402 (1986).
16) Gill, J. T. III: Computational Complexity of Probabilistic Turing Machines, SIAM J. Comp. 6, pp. 675-695 (1977).
17) Goldwasser, S.-Kilian, J.: Almost All Primes Can Be Quickly Certified, Proc. 18th Ann. ACM Symp. Theory Comp., pp. 316-329 (1986).
18) Johnson, D. S.: The NP-Completeness Column: An Ongoing Guide, J. of Algorithms, 7, pp. 584-601 (1986).
19) Miller, G. L.: Riemann's Hypothesis and Tests for Primality, J. Comp. Syst. Sci., 13, pp. 300-317 (1976).
20) Pollard, J. M.: A Monte-Carlo Method for Factorization, BIT 15, pp. 331-334 (1975).
21) Pratt, V. R.: Every Prime Has a Succinct Certificate, SIAM J. Comput., 4, pp. 214-220 (1975).
22) Rabin, M. O.: Probabilistic Algorithm for Testing Primality, J. Number Theory, 12, pp. 128-138 (1980).
23) Schoof, R.: Quadratic Fields and Factorization, Lenstra-Tijdeman ed., Computational Methods in Number Theory, Amsterdam Math. Cent. (1984).
24) Schoof, R.: Elliptic Curves Over Finite Fields and the Computation of Square Roots Mod p. Math, Comp. 44, pp. 483-494 (1985).
25) Shanks, D.: Class Number; A Theory of Factorization and Genera, Proc. Symp. in Pure Math., 20, Amer. Math. Soc., Providence, pp. 415-440 (1971).
26) Solovay, R.-Strassen, V.: A Fast Test for Primality, SIAM J, Comput., 6, pp. 84-85 (1977).
27) Yao, A.: Theory and Application of Trapdoor Functions, Proc. 23rd Ann. Symp. Found. Comput. Sci., pp. 80-91 (1982).

(平成4年9月14日受付)



一松 信 (正会員)

1926年生．1947年東京大学理学部数学科卒業．1954年理学博士（東大，旧制）．立教大学助教授・東京大学助教授・立教大学教授を経て1969年京都大学数理解析研究所教授．1989年同上定年退職・京都大学名誉教授，東京電機大学教授（理工・情報）．著書：数値解析（朝倉）他多数．テーマ：数式処理と数学研究への応用．日本数学会，ソフトウェア科学会各会員．日本数学教育学会顧問．