高校生のための現代数学講座
「素数の世界」
講義 (2)　坪井　俊

東京大学
玉原国際セミナーハウス
2009 年 7 月 25 日

「最大公約数」

1　２つの自然数の最大公約数

- 「定義」２つの自然数 m, n に対し、m、n の両方を割り切る自然数 a を、m, n の公約数という。
- ２つの自然数 m, n の公約数のなかで、最大のものを最大公約数といい、$\mathrm{GCD}(m, n)$ あるいは、単に (m, n) と書く。英語 greatest common divisor の頭文字である。
- $m > n$ のとき、m と n の公約数は、$m - n$ の約数であり、n と $m - n$ の公約数は、m, n の約数である。従って、$\mathrm{GCD}(m, n) = \mathrm{GCD}(n, m - n)$.
- $m > n$ のとき、m を n で割ると q 余り r となるとする。すなわち、$m = n \times q + r$ $(0 \leqq r < n)$ とする。このとき、m と n の公約数は、r の約数であり、n と r の公約数は、m, n の約数である。従って、$\mathrm{GCD}(m, n) = \mathrm{GCD}(n, r)$.

2　ユークリッドの互除法

- 割り算の余りと公約数の関係は、最大公約数を求める手順を与える。この手順は、計算機にプログラムとして組めるものであり、アルゴリズムと呼ばれる。

(1) 自然数 m, n に対して、$n_0 = m$, $n_1 = n$ とおく。

(2) 自然数 k に対し、n_{k-1}, n_k が与えられ、$n_k \neq 0$ の時、$n_{k-1} = n_k \times r_k + n_{k+1}$ $(0 \leqq n_{k+1} < n_k)$ となる n_{k+1} を割り算で計算する。

(3) 最初に $n_{k+1} = 0$ になったとき、すなわち、n_{k-1} が n_k で割り切れるとき、n_k が m, n の最大公約数である。

- ここで大切なのは、$n_1 > n_2 > n_3 > \ldots$ と減少していくので、ある k で、必ず $n_{k+1} = 0$ となることである。
- 例えば、次のように計算され、2009 と 707 の最大公約数は 7 であることが分かる。

$$\begin{array}{rl} 2009 \div 707 = 2 \text{ 余り } & 595 \\ 707 \div 595 = 1 \text{ 余り } & 112 \\ 595 \div 112 = 5 \text{ 余り } & 35 \\ 112 \div 35 = 3 \text{ 余り } & 7 \\ 35 \div 7 = 5 \text{ 余り } & 0 \end{array} \qquad \begin{array}{rl} 2009 = 707 \times 2 + & 595 \\ 707 = 595 \times 1 + & 112 \\ 595 = 112 \times 5 + & 35 \\ 112 = 35 \times 3 + & 7 \\ 35 = 7 \times 5 & \end{array}$$

- 筆算を次のように書くと見直しやすい。筆算の書き方はいろいろと工夫できる。

1	707	2009	2
	595	1414	
3	112	595	5
	105	560	
	7	35	5
		35	
		0	

- 十進BASICのプログラムで書くと次のようになる。

```
! ユークリッドの互除法によりGCD(a,b)を求める                    コメント
INPUT PROMPT  "a > b > 0となる整数":a,b    2つの整数の入力を求める
PRINT                                                          改行
10 LET  q=INT(a/b)                          分数a/bの整数部分をqとする
   LET  r=MOD(a,b)                          a÷bの余りをrとする
   IF r =0 THEN GOTO 30 ELSE GOTO 20
                        r = 0ならば、行番号30へ、そうでなければ行番号20へ
20 PRINT a;"=";b;"×";q;"+";r                    a=b × q+r と書く
   LET a=b                                      aの値をbに取り換える
   LET b=r                                      bの値をrに取り換える
   GOTO 10                                                行番号10へ
30 PRINT a;"＝";b;"×";q                          a=b × q と書く
END                                                        プログラム終了
```

- 十進BASICはフリーのソフトウェアで、BASICのプログラムをＰＣ上で実行するものです。http://hp.vector.co.jp/authors/VA008683/からダウンロードできます。

3　2つの自然数の和や差で書ける整数

- ユークリッドの互除法で最大公約数を求めることができることから、最大公約数 $\mathrm{GCD}(m,n)$ に対して、整数 a, b で、$\mathrm{GCD}(m,n) = am + bn$ と書くものが存在することが分かる。

- 例えば、$\mathrm{GCD}(2009, 707) = 7$ の場合には、次のように計算できる。

$$\begin{aligned} 7 &= 112 - 3 \times 35 \\ &= 112 - 3 \times (595 - 5 \times 112) = -3 \times 595 + 16 \times 112 \\ &= -3 \times 595 + 16 \times (707 - 1 \times 595) = 16 \times 707 - 19 \times 595 \\ &= 16 \times 707 - 19 \times (2009 - 2 \times 707) = -19 \times 2009 + 54 \times 707 \end{aligned}$$

- 一方、順に計算するならば次のようにする。これは、プログラムしやすい。

$$\begin{aligned} 2009 &= 707 \times 2 + 595 \\ 707 &= 595 \times 1 + 112 \\ 595 &= 112 \times 5 + 35 \\ 112 &= 35 \times 3 + 7 \\ 35 &= 7 \times 5 \end{aligned} \qquad \begin{aligned} 595 &= 1 \times 2009 - 2 \times 707 \\ 112 &= -1 \times 2009 + 3 \times 707 \\ 35 &= 6 \times 2009 - 17 \times 707 \\ 7 &= -19 \times 2009 + 54 \times 707 \end{aligned}$$

- a, b を整数として、$am + bn$ の形に書かれる整数は、$\mathrm{GCD}(m,n)$ の倍数全体に一致する。

4　割り算による素数の判定

- ある自然数 n が素数であることを調べるためには、n が n 未満の自然数で割り切れないことを確かめればよい。
- もしも n が m で割り切れて、2 つの自然数の積 $n = \ell \times m$ と書かれれば、ℓ または m の一方は、n の平方根以下である。従って、素数であることを調べるためには、n が $\sqrt{n}$ 以下の自然数で割り切れないことを確かめればよい。
- 2 進数で書かれている m, n の割り算は、n が d 桁とすると、「m の上 d 桁」$\leqq n$ ならば、商に 1 を立てて、n を引く。「m の上 d 桁」$< n$ ならば、m の上 $d+1$ 桁に対して、商に 1 を立てて、n を引く。というプロセスで行われる。
- これは、2 進数の n 桁の比較および計算を、ほぼ（m の桁数 $-n$ の桁数）の回数繰り返していることになる。
- 例えば、$595 = 2^9 + 2^6 + 2^4 + 2^1 + 1$ で 2 進数の表記は 1001010011 である。$112 = 2^6 + 2^5 + 2^4$ で 2 進法の表記は 1110000 である。$595 \div 112 = 5$ 余り 35 の計算は次のようになる。

$$
\begin{array}{r}
101 \\
1110000\,)\overline{1001010011} \\
\underline{1110000} \\
10010011 \\
\underline{1110000} \\
100011
\end{array}
$$

- 2 進数の表記は、10^3 と $1024 = 2^{10}$ がほぼ等しいから、１０進数の表記の 10/3 倍の桁数になる。
- 全体で素数の判定のためには、およそ $\displaystyle\sum_{n=2}^{\sqrt{m}} \log_2 n \times (\log_2 m - \log_2 n)$ に比例した計算の量が必要である。ここで、$\log_2 n$ は n の 2 進法による桁数を d とすると、$2^{d-1} \leqq n \leqq 2^d - 1$ だから、$d - 1 \leqq \log_2 n < d$ を満たしている。
- 仮に m が 2 進法で 2ℓ 桁とすると、$\sqrt{m}$ は 2 進法で ℓ 桁である。k 桁の 2 進数は 2^{k-1} 個ある。この場合、和は $\displaystyle\sum_{k=1}^{\ell} 2^{k-1} \times k \times (2\ell - k)$ に比例した計算の量となる。これは $\ell \displaystyle\sum_{k=1}^{\ell} 2^{k-1} \times k$ 以上 $2\ell \displaystyle\sum_{k=1}^{\ell} 2^{k-1} \times k$ 以下である。$\displaystyle\sum_{k=1}^{\ell} 2^{k-1} \times k = \ell(1 + (\ell - 1)2^{\ell})$ で、計算量は 2 進数の桁数 2ℓ に対し、$\ell(\ell - 1)2^{\ell}$ 以上である。
- ここで、$\displaystyle\sum_{k=0}^{\ell} x^k = \frac{1 - x^{\ell+1}}{1 - x}$ だから、$\displaystyle\sum_{k=1}^{\ell} k x^{k-1} = \frac{1}{(1-x)^2}(1 - x^{\ell+1}) - \frac{\ell + 1}{1 - x} x^{\ell}$ であり、$x = 2$ として、$\displaystyle\sum_{k=0}^{\ell} k 2^{k-1} = 1 - 2^{\ell+1} + (\ell + 1)2^{\ell} = 1 + (\ell - 1)2^{\ell}$ となることを用いた。

- 結局、$2^{2\ell-1}$ 以上 $2^{2\ell}$ 未満の自然数が素数かどうか調べるために、ほぼ $\ell(\ell-1)2^{\ell}$ 回の定数倍の計算が必要になる。従ってそれだけ時間がかかる。

- 2 進法で 10 桁 (10 進法で 3 桁) の自然数に対する時間が T ならば、2 進法で 20 桁 (10 進法で 6 桁) の自然数に対する計算にはほぼ、$2^{12}T$ かかる。2 進法で 40 桁 (10 進法で 12 桁) に対しては、$2^{34}T$, 2 進法で 80 桁 (10 進法で 24 桁) に対しては、$2^{76}T$, 2 進法で 160 桁 (10 進法で 48 桁) に対しては、$2^{158}T$ となる。つまり、10 進法 6 桁で 4000 倍、12 桁で 10^8 倍、24 桁で 10^{22} 倍、48 桁で 10^{49} 倍かかることになる。

- 実際には、この PC 上での十進 BASIC のプログラムでは 10 桁 (9999990017) 0.18 秒、12 桁 (999999990047) 1.25 秒、14 桁 (99999999990001) 7.71 秒、16 桁 (9999999999990019) 70.89 秒、18 桁 (999999999999990029) 716.49 秒、20 桁 (99999999999999990167) 7919.54 秒 （40000 倍）である。

- この計算は、素数であるかどうかを調べるだけではなく、$\sqrt{m}$ までの**約数をすべて求めるもの**である。

- 素数であることだけを判定するならば、桁数のべき乗で計算量が評価できるアルゴリズムが２１世紀の初めに見つかっている。「素数の判定」というキーワードで検索すると情報が見つかる。

5 ユークリッドの互除法の計算量

- ２つの自然数の最大公約数を見つけるユークリッドの互除法は、桁数に対し、その何倍かの計算量で実行できる。

- このことは、次のように説明できる。

(1) ２進法で書かれた m, n の桁数を ℓ_m, ℓ_n とすると、m を n で割る割り算には、$(\ell_m - \ell_n)\ell_n$ 回の計算が必要である。

(2) その次には、ℓ_n 桁以下のものが２つのこる。これの割り算は、${\ell_n}^2$ 回以下の計算である。

(3) ２回の割り算で得られた２つの数の小さいほうの桁数は $\ell_n - 1$ 以下である。

(4) 従って、$2\ell_n$ 回以内に互除法は必ず終わる。

- 十進法で n の桁数が、k_n ならば、２進法では n の桁数はほぼ $\dfrac{10}{3}k_n$ で、互除法は、$\dfrac{20}{3}k_n$ 回で終了することになる。実際には、多くて $5k_n$ 回程度である。

- ユークリッドの互除法は桁数が大きくても、短時間でできる。合同式の計算も短時間でできることになる。