ラメの定理 【ユークリッドの互除法の計算回数】

a, b を $a>b$ であるような正の整数とし，a は b で割り切れないとする。このとき

$$\begin{aligned}
&a \text{ を } b \text{ で割ったときの商を } q_1\text{，余りを } r_1\\
&b \text{ を } r_1 \text{ で割ったときの商を } q_2\text{，余りを } r_2\\
&r_1 \text{ を } r_2 \text{ で割ったときの商を } q_3\text{，余りを } r_3\\
&r_2 \text{ を } r_3 \text{ で割ったときの商を } q_4\text{，余りを } r_4\\
&\qquad\vdots\\
&r_{N-3} \text{ を } r_{N-2} \text{ で割ったときの商を } q_{N-1}\text{，余りを } r_{N-1}\\
&r_{N-2} \text{ を } r_{N-1} \text{ で割ったときの商を } q_N\text{，余りを } r_N(=0)
\end{aligned}$$

とする。

また，$\begin{cases} F_{n+2}=F_{n+1}+F_n \\ F_1=F_2=1 \end{cases}$ で定まる数列 $\{F_n\}$ を考える。

(1) $N=2$ のとき，$r_1 \geqq F_2$ を示せ。

(2) $N \geqq 3$ とする。$k=2, 3, \cdots, N$ に対して，$r_{N+1-k} \geqq F_k$ を示せ。

(3) $b \geqq F_{N+1}$ であることを示せ。

(4) φ を $x^2-x-1=0$ の正の解，すなわち $\varphi=\dfrac{1+\sqrt{5}}{2}$ とするとき，$F_{n+1} \geqq \varphi^{n-1}$ であることを示せ。

(5) b が M 桁であるとき，$N \leqq 5M$ であることを示せ。

【戦略】

(1) **基本的に余りを作っていくアルゴリズムなので，$r_1, r_2, \cdots$ は $r_1>r_2>\cdots>r_{N-1}>r_N(=0)$ という単調減少数列です。**

$N=2$ とは $a=bq_1+r_1$，$b=r_1q_2$ $(r_2=0)$ と，2 回の割り算で互除法のプロセスが完了するということです。

$r_1>r_2(=0)$ なので，$r_1 \geqq 1(=F_2)$ となり，証明完了です。

(2) **漸化式が与えられているので，手なりに数学的帰納法を選択したいところです。**

k についての数学的帰納法だということに気を付けてください。

(3) **$b=r_1q_2+r_2$ であり，商の q_2 は 1以上なので，$b \geqq r_1+r_2$ です。**

ここで，(2) の不等式の出番で，(2) の不等式は (1)で考えた $N=2$ のときも整合性が取れています。

よって $r_{N+1-N} \geqq F_N$，$r_{N+1-(N-1)} \geqq F_{N-1}$ $(N=2, 3, \cdots)$

すなわち $r_1 \geqq F_N$，$r_2 \geqq F_{n-1}$ を得るため，辺々加えれば解決です。

(4) **添え字に関する範囲のうるさい今までの話と違って単純な不等式証明であり，基本的な帰納法の運用で片付きます。**

(5) **今まで証明した不等式と，b の桁数に関する条件から**

$$\varphi^{N-1} \leqq F_{N+1} \leqq b < 10^M$$

と言えます。

$\varphi^{5(N-1)} \leqq {F_{N+1}}^5 \leqq b^5 < 10^{5M}$ なので，$10<\varphi^5$ が言えれば解決でしょう。

$10^{N-1}<\varphi^{5(N-1)}<10^{5M}$ なので，$N-1<5M$ となりますから。

【解答】

$$\begin{aligned}
a&=bq_1+r_1\\
b&=r_1q_2+r_2\\
r_1&=r_2q_3+r_3\\
r_2&=r_3q_4+r_4\\
&\vdots\\
r_{k-2}&=r_{k-1}q_k+r_k\\
&\vdots\\
r_{N-3}&=r_{N-2}q_{N-1}+r_{N-1}\\
r_{N-2}&=r_{N-1}q_N+r_N \quad (r_{N-2}=r_{N-1}q_N)
\end{aligned}$$

まず，$r_1, r_2, \cdots$ は単調減少の整数の数列であり

$$r_1>r_2>r_3>\cdots>r_k>\cdots>r_{N-2}>r_{N-1}>r_N(=0) \quad \cdots ①$$

(1) $N=2$ のとき

① より $r_1>r_2(=0)$ なので，$r_1 \geqq 1(=F_2)$

ゆえに，$r_1 \geqq F_2$ が成立する。

(2) $k=2, 3, \cdots, N$ に対して，$r_{N+1-k} \geqq F_k$ $\cdots(*)$ であることを k についての数学的帰納法で証明する。

(i) $k=2, 3$ のとき

① より，$r_{N-1} \geqq 1$，すなわち $r_{N+1-2} \geqq F_2$ であり $k=2$ のとき $(*)$ は成立する。

また，$r_{N-2}>r_{N-1}>r_N(=0)$ であるため，$r_{N-2} \geqq 2$

すなわち，$r_{N+1-3} \geqq F_3$ であり，$k=3$ のとき $(*)$ は成立する。

(ii) $k=\ell$，$\ell+1$ のときに $\begin{cases} r_{N+1-\ell} \geqq F_\ell \\ r_{N+1-(\ell+1)} \geqq F_{\ell+1} \end{cases}$ であると仮定する。

すなわち，$\begin{cases} r_{N+1-\ell} \geqq F_\ell \\ r_{N-\ell} \geqq F_{\ell+1} \end{cases}$ $\cdots ②$

このとき，互除法のアルゴリズムによる数列 $\{r_n\}$ の定め方から

$$r_{N-(\ell+1)}=r_{N-\ell}q_{N+1-\ell}+r_{N+1-\ell}$$

商 $q_{N+1-\ell}$ は 1 以上であるから，$r_{N-(\ell+1)} \geqq r_{N-\ell}+r_{N+1-\ell}$

② より，$r_{N-(\ell+1)} \geqq F_\ell+F_{\ell+1}$

これより，$r_{N+1-(\ell+2)} \geqq F_{\ell+2}$ が言え，$k=\ell+2$ のときも $(*)$ は成立する。

(i), (ii) から，$k=2, 3, \cdots, N$ に対して，$r_{N+1-k} \geqq F_k$ が成立する。

(3) $b=r_1q_2+r_2$ で，商の q_2 は 1 以上であるため，$b\geqq r_1+r_2$

(1)，(2) から，$r_{N+1-N}\geqq F_N$，$r_{N+1-(N-1)}\geqq F_{N-1}$ $(N=2,3,\cdots)$

すなわち，$r_1\geqq F_N$，$r_2\geqq F_{N-1}$ なので

$$\begin{aligned} b&\geqq r_1+r_2\\ &\geqq F_N+F_{N-1}\\ &=F_{N+1}\end{aligned}$$

となり，示された。

(4) $F_{n+1}\geqq\varphi^{n-1}$ …(∗)′ であることを n についての数学的帰納法で示す。

(Ⅰ) $n=1,2$ のとき

$F_2-\varphi^0=1-1=0$ で，$n=1$ のとき (∗)′ は成立する。

$$\begin{aligned} F_3-\varphi&=2-\frac{1+\sqrt{5}}{2}\\ &=\frac{3-\sqrt{5}}{2}>0\end{aligned}$$ で，$n=2$ のとき (∗)′ は成立する。

(Ⅱ) $n=K$，$K+1$ $(K=1,2,\cdots)$ のとき

$F_{K+1}\geqq\varphi^{K-1}$，$F_{K+2}\geqq\varphi^K$ が成立すると仮定する。

$$\begin{aligned} F_{k+3}&=F_{K+1}+F_{K+2}\\ &\geqq\varphi^{K-1}+\varphi^K\\ &=\varphi^{K-1}(\varphi+1)\\ &=\varphi^{K-1}\cdot\varphi^2\ (\because\ \varphi\text{ は }\varphi^2-\varphi-1=0\text{ を満たしている})\\ &=\varphi^{K+1}\end{aligned}$$

となり，$n=K+2$ のときも (∗)′ は成立する。

(Ⅰ)，(Ⅱ) から，$n=1,2,\cdots$ に対して，$F_{n+1}\geqq\varphi^{n-1}$ が成立する。

(5) b は M 桁なので，$b<10^M$ が成立する。

(3)，(4) も併せて考えると，$\varphi^{N-1}\leqq F_{N+1}\leqq b<10^M$ …(☆) が成立する。

ここで，$\varphi^5>10$，すなわち $\left(\frac{1+\sqrt{5}}{2}\right)^5>10$ である。

これを示すには，$(1+\sqrt{5})^5>2^5\cdot10\ (=320)$ を示せばよい。

$$\begin{aligned} (1+\sqrt{5})^5&=1+{}_5\mathrm{C}_1(\sqrt{5})+{}_5\mathrm{C}_2(\sqrt{5})^2+{}_5\mathrm{C}_3(\sqrt{5})^3+{}_5\mathrm{C}_4(\sqrt{5})^4+(\sqrt{5})^5\\ &=1+5\sqrt{5}+50+50\sqrt{5}+125+25\sqrt{5}\\ &=176+80\sqrt{5}\\ &=176+\sqrt{32000}\\ &>176+\sqrt{20736}\\ &=176+\sqrt{144^2}\\ &=176+144\\ &=320\end{aligned}$$

これより，$\varphi^5>10$ であり，$\varphi^{5(N-1)}>10^{N-1}$ …(★)

(☆)，(★) より，$10^{N-1}<\varphi^{5(N-1)}<10^{5M}$

特に，$N-1<5M$ であるため，$N-1<5M$，すなわち $N<5M+1$

N は整数であるから，$N\leqq5M$ である。

【総括】

N というのはユークリッドの互除法を用いて最大公約数を求める際の最大計算回数です。

例えば，b が 10 桁であれば，高々 50 回のアルゴリズムで互除法のプロセスが終わることを意味します。

3847563923 のような 10 桁の数で素因数を探すとなると

$\sqrt{3847563923}\fallingdotseq62028.**\cdots$ 以下の素数で素因数を探すことになります。

60000 までの中で考えても素数は 6057 個ありますから，これらの素数が素因数になっているかどうかを判定するよりも，高々 50 回のアルゴリズムで最大公約数を求める方が遥かに効率的であることが分かります。

なお，N が一番大きくなる (互除法のプロセスが最も長引く) ケースを考えてみます。

$r_1>r_2>\cdots\cdots>r_N\ (=0)$ ですから，数列 $\{r_n\}$ が中々減少しないときを考えるわけです。(早く減っちゃったら，それだけ早く 0 に近づいちゃう)

商が 1 となるときが余りの減少スピードが遅くなりますね。

例えば，$13=3\cdot4+1$ というのは，13 から 3 をできる限り取り除いて (4 回) もうこれ以上 3 を取り除けず残ったものが「余り」です。

余りが減少しないようにするわけですから，取り除く回数 (商) が小さいほうが望ましいわけです。

つまり，一番互除法の手数が大きくなるケースというのは
互除法の商が全て 1 となるような場合

$$\begin{aligned} a&=b+r_1\\ b&=r_1+r_2\\ r_1&=r_2+r_3\\ r_2&=r_3+r_4\\ &\vdots\end{aligned}$$

すなわち，$r_{k-2}=r_{k-1}+r_k$ のときであることが想像がつくわけです。

言ってみれば，フィボナッチ数列が逆から並んでいるような感じでしょうか。

フィボナッチ数列を並べてみると

$$1,1,2,3,5,8,13,21,34,55,89,144,233,\cdots$$

です。

例えば，233 と 144 に対して互除法のアルゴリズムをやってみると

（1，1，2，3，5，8，13，21，34，55，89，144，233）

$$233=144\cdot1+89$$
$$144=89\cdot1+55$$
$$89=55\cdot1+34$$
$$55=34\cdot1+21$$
$$34=21\cdot1+13$$
$$21=13\cdot1+8$$
$$13=8\cdot1+5$$
$$8=5\cdot1+3$$
$$5=3\cdot1+2$$
$$3=2\cdot1+1$$
$$2=1\cdot2$$

ここを見ながら計算すればよいでしょう。

フィボナッチ数列を逆順に見ているという感覚をもつと，(2) で示した $r_{N+1-k}\geqq F_k$ という不等式の「添え字の逆転性」の意味も納得でしょう。

ラメの定理

a，b を $a>b$ なる自然数として，小さいほうの b の桁数を M とするとき，ユークリッドの互除法により最大公約数を求めるアルゴリズムの回数は

高々 $5M$ 回