

進数表現における、数とその各桁の総和の関係

環境情報学部4年 石原和音

学籍番号 70050607

t00069ki@sfc.keio.ac.jp

kazuto@kz-soft.com

2006/6/14 14:38

『ある数 x の各桁の総和が 3 で割り切れるとき、 x 自身も 3 で割り切れる』ことはよく知られている。これは 10 進数においてのみ成立する。では、一般に p 進数においてどのようなことが言えるだろうか。

本論では、ある数 x の p 進数における各桁の総和を $q_0 \in \{q \in \mathbb{N} \mid q \cdot n = p - 1 \wedge n \in \mathbb{N}\}$ で割った余りは、 x そのものを q_0 で割った余りに一致することを証明する。

I. $\forall m \in \mathbb{N}. \forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ を示す

1. まず、 $m=0$ のとき $a \cdot p^m = a \cdot p^0 = a \cdot 1 = a$ より $a \cdot p^m \equiv a \pmod{q}$ は自明。

$\forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ が成立。

2. $m=s$ のとき $\forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ が成立すると仮定する。すなわち $\forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^s \equiv a \pmod{q}$

いま、 $a \cdot p^{s+1}$ について考えると、

$$\begin{aligned} a \cdot p^{s+1} &= a \cdot p^s \cdot p \\ &= (a \cdot p^s \cdot 1) + (a \cdot p^s \cdot (p - 1)) \\ &= a \cdot p^s + q(a \cdot p^s \cdot n) \end{aligned}$$

ただし、 $q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1$

$(a \cdot p^s \cdot n)$ は自然数なので、 $q(a \cdot p^s \cdot n)$ は q の倍数。よって $a \cdot p^{s+1} \equiv a \cdot p^s \pmod{q}$

いま、仮定より $a \cdot p^s \equiv a \pmod{q}$ だから、 $a \cdot p^{s+1} \equiv a \pmod{q}$

$m=s+1$ のときも

$\forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ が成立。

3. 数学的帰納法により、1. 2. から全ての $m \in \mathbb{N}$ について

$\forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p - 1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ が成立。

II. p 進数での表現

x は p 進数でどのように表記できるだろうか。

$$x = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \cdots + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0$$

このように、 x は $n+1$ 桁の p 進数で“ $a_n a_{n-1} \cdots a_1 a_0$ ”と表す。

前節で、 $\forall m \in \mathbb{N}. \forall p \in \mathbb{N}. \quad q \in \mathbb{N} \wedge n \in \mathbb{N} \wedge qn = p-1 \Rightarrow a \cdot p^m \equiv a \pmod{q}$ が言えた。

そこで、 $x = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \cdots + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0$ の右辺に注目すると、

前節の結果より $a_m \cdot p^m \equiv a_m \pmod{q}$ であるから、

$$\begin{aligned} x & \\ & \equiv a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \cdots + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0 \\ & \equiv a_n + a_{n-1} + \cdots + a_2 + a_1 + a_0 \pmod{q} \end{aligned}$$

$$\because \alpha_1 \equiv \alpha_2 \wedge \beta_1 \equiv \beta_2 \Rightarrow \alpha_1 + \beta_1 \equiv \alpha_2 + \beta_2 \pmod{q}$$

よって、 x を q で割った余りは、各桁の総和を q で割った余りに等しい。(証明終)

IV. 応用

上記の性質は以下のようなことに応用できる。

手計算を簡単にする

これはよく知られた性質であるが、10進数を $3,9$ で割った余りを求めたいとき、各桁の総和を $3,9$ で割った余りを求めればよい。10進数が割り切れるかどうか調べるためには、余りが0であるかどうかを調べればよい。 $3,9$ で成立するのは、 $10-1$ の約数だからである。

同様に、16進数を $3,5,15$ ⁽¹⁾で割った余りを求めたいとき、各桁の総和を $3,5,15$ で割った余りを求めればよい。

ちなみに1で割った余りは必ず0であるので、意味がない。

計算機における多倍長演算での応用

多倍長演算とは、計算機において、とても大きな数を扱いたいとき、それをCPUで直接扱える(8bit, 16bit, 32bit, 64bitなどの)細かい単位に分けて計算を行う算法⁽²⁾である。多倍長演算は、進数表現による計算の一種であると考えられる。たとえば、8bitを一桁とみて、 $2^8=256$ 進数として計算している、と考えられる。同様に64bitを単位とした多倍長演算ならば、64bitを一桁と見て $2^{64}=18446744073709551616$ 進数として計算している、と考えられる。この考え方は、我々が10進数で筆算を行うときと本質的に同じである。では、多倍長演算で、本論に述べる性質はどのように応用できるだろうか。

たとえば $2^8=256$ 進数の多倍長演算において、 $q \in \{1, 3, 5, 15, 17, 51, 85, 255\}$ ⁽³⁾で巨大な数 x を割った余りが欲しいとき、 x を直接割って余りを求めるのではなく、各桁の合計を q で割った余りを使えばよい。

同様に、たとえば 2^{64} 進数の多倍長演算において、

$$q \in \{1, 3, 5, 15, 17, 51, 85, 255, 257, \dots, 6148914691236517205, 18446744073709551615\}$$
⁽⁴⁾

などで x を割った余りが欲しいとき、各桁の合計を q で割った余りを使えばよい。

特に、受動的にこの性質を利用するのではなく、何らかの数 r で剰余をとる必要がある算法において、積極的に $r=q$ 、 $q \in \{\alpha | \alpha \text{は } p-1 \text{の約数}\}$ を選ぶとよい。

また、 x が非常に大きいときは、各桁の剰余(余り)の総和の剰余をとれば、更に効率的になるだろう。

(1) $16-1=15$ の約数である。

(2) アルゴリズム

(3) $2^8-1=255$ の約数である。

(4) $2^{64}-1$ の約数である。

性質を逆に利用する

$p-1$ が素数であれば、逆に p 進数ではこのような剰余の効率的な計算が出来ないということになる。暗号・認証などに対する、この性質を使った攻撃を避け、計算の困難さを維持することに寄与するかもしれない。現代では計算機において暗号・認証が行われることを考えれば、先ほどの多倍長演算と関連して考え、 $p-1=2^u-1$ が素数⁽⁵⁾となるように

$u \in \{2, \dots, 61, 89, 107, 127, 521, 607, 1279, \dots\}$ を選び、 u bit の多倍長演算を行うことになるだろう。

(5) 2^u-1 ($u \in \mathbb{N}$) と表せる素数をメルセンヌ素数という。cf.

<http://ja.wikipedia.org/wiki/%E3%83%A1%E3%83%AB%E3%82%BB%E3%83%B3%E3%83%8C%E6%95%B0>