mod演算

Iwa

2021年7月17日

1　modの定義

整数x, yをnで割った余りが等しいとき，xとyはnを法として合同であるといい，

$$x \equiv y \pmod{n}$$

とかき，これを合同式と呼ぶ．これは適当な整数kを用いて

$$x = nk + y$$

とかくのと同値である．この場合特にyを余りと呼ぶ．
mod演算は余りのみに注目し，法nの倍数部分は無視する演算と言える．プログラミング言語C++では演算子%でmodを表現する．つまり$x \equiv y \pmod{n}$は

$$x\%n == y\%n$$

とかく．
mod演算の例

$$\begin{aligned} 13 &\equiv 23 \pmod{5} \\ &\equiv 18 \pmod{5} \\ &\equiv 3 \pmod{5} \\ &\equiv -2 \pmod{5} \end{aligned}$$

最後の負の数は

$$-2 = 5 \cdot (-1) + 3 \equiv 3 \pmod{5}$$

ということである．
modで負の数を扱うこともできるが，余りは一般的に，剰余類で定義されるように$0 \leq r \leq n-1$とするのが多い．

2　加法　減法　乗法　累乗

除算以外の演算，加法　減法　乗法　累乗について普通の式と同様に演算することができる．加法　減法　乗法について以下の合同式が成り立つ．
$a \equiv b,\ c \equiv d \pmod{n}$ のとき

$$a \pm c \equiv b \pm d \pmod{n}$$

$$ac \equiv bd \pmod{n}$$

これらの証明は $x = nk + y$ をそれぞれ代入して計算することで確かめられる．
また，上式は以下の式と同値

$$(a \pm c) \pmod{n} = \{a \pmod{n}\} \pm \{c \pmod{n}\}$$

$$(a * c) \pmod{n} = \{a \pmod{n}\} * \{c \pmod{n}\}$$

C++で書くと，

$$(a \pm c)\%n == a\%n \pm c\%n$$

$$(a * c)\%n == (a\%n) * (c\%n)$$

累乗について，乗法の式を繰り返し用いることで以下の合同式が成り立つ．
整数 m を用いて $a \equiv b \pmod{n}$ のとき

$$a^m \equiv b^m \pmod{n}$$

C++では

$$a^m == (a\%n)^m$$

以上で挙げた性質は，最終的な答えが出る前に $10^9 + 7$ で割っても，正しい解答の余りが得られることを保証している．

3　除法と逆元

3.1　除法

割る数 m と法 n が互いに素である時,

$$am \equiv bm \Longleftrightarrow a \equiv b \pmod{n}$$

が成り立つ．以下では割った後整数にならない場合について考える．
b と p が互いの素である時だけ,

$$x \equiv a \div b \pmod{p}$$

が法 p で一意に定まる．言い換えればこれは両辺に b をかけて

$$bx \equiv a \pmod{p}$$

とかいて，x が法 p で一意に定まるということと同じ．
具体例を示す．$6 \div 10 \pmod{7}$ を計算したい．これを x とおいて

$$6 \div 10 \equiv x \pmod{7}$$

両辺に 10 をかけることで

$$6 \equiv 10x \pmod{7} \tag{1}$$

を満たすような x を見つける問題に帰着する．
例えば $x = 2$ としてみると $20 \equiv 6 \pmod{7}$ だから確かに (1) 式を満たす．よって

$$6 \div 10 \equiv 2 \pmod{7}$$

と求まる．
つまり，一般に $x = a \div b \pmod{n}$ を求めるとは

$$bx \equiv a \pmod{p} \tag{2}$$

を満たす x を探すことになる．

3.2　逆元

上記のような x を求める時，以下の性質を満たす数があるとよい．

$$b^{-1}b \equiv 1 \pmod{p} \tag{3}$$

このような (3) 式を満たす数のことを b の逆元といい b^{-1} で表す．このような逆元は b と p が互いに素である時，存在する．
b^{-1} が存在すれば (2) 式の両辺にそれをかけて

$$x \equiv b^{-1}a \pmod{p}$$

と x を表すことができる．b^{-1} という表記は，$\div b$ や分母の b と同一視することができ，例えば

$$6 \div 10 \equiv 6 \cdot 10^{-1} \pmod{n}$$

$$_nC_k = \frac{n!}{k!(n-k)!} \equiv n!(k!)^{-1}(n-k)^{-1} \pmod{p}$$

このように実際に割り算をせずに逆元を掛けることで，除算を表現する．逆元を求める手法はフェルマーの小定理，ユークリッドの互除法を利用したものがある．ただし，フェルマーの小定理の方法では法が素数のときしか求められない．

3.3 フェルマーの小定理 Fermat の小定理-逆元を求める手法 1

フェルマーの小定理
p を素数，a を p の倍数でないとして

$$a^{p-1} \equiv 1 \pmod{p}$$

が成り立つ．
左辺を

$$a^{p-1} = a \cdot a^{p-2}$$

と見れば a^{p-2} が a の逆元になっている．そのまま計算すると $O(p)$ かかるが，二分累乗法によって計算量が $O(\log\ p)$ になる．
また $r^n \equiv 1 \pmod{p}$ となるとなる最小の正の整数 n を法 p における p の位数と呼ぶ．
素数 p に対して位数が $p-1$ であるような元 r を p の原始根と呼ぶ．
素数 p には原始根 r が必ず存在する．(原始根の存在定理)
素数 p に対する原始根を r とすると，$r, r^2, \cdots, r^{p-1}$ を p で割った余りは全て異なり 1 から $p-1$ までを一巡する．

3.4 ユークリッドの互除法ー逆元を求める手法 2

以下の記号/はプログラミングにおける意味で，整数同士の場合小数点以下が切り捨てられるものである．任意の整数 a は，法 n で

$$x = n * (x/n) + x\%n \tag{4}$$

これはちょうど

$$x = nk + r$$

に対応している．(4) 式を変形する．

$$\begin{aligned} 0 &\equiv n * (x/n) + x\%n \pmod{x} \\ 0 &\equiv (x/n) + x\%n \cdot n^{-1} \pmod{x} \\ x\%n * n^{-1} &\equiv -(x/n) \pmod{x} \\ n^{-1} &\equiv -(x/n) * (x\%n)^{-1} \pmod{x} \end{aligned}$$

4　競プロ問題での利用

4.1　負の数の mod

競プロにおいて余り r を正の整数で答える問題が多いが，C++の演算%は負の数の余りを扱う時，負の数を返す事があるため不都合な時がある．実際に

```
1  int a=-2;
2  int b=-8;
3  int n=5;
4  cout<< a%n <<endl;
5  cout<< b%n <<endl;
```

というコードの実行結果は

$$-2$$

$$-3$$

となり，正の整数でない．これは a が負のとき $a\%n$ を $a\%n + n$ とすることで対策できる．

4.2　互いに素

除算や逆元では割る数と法が互いに素であることが重要であった．必ずしも素数である必要はないが，もし素数ならその素数自体の倍数以外と，その素数とは互いに素となる．$10^9 + 7$ は素数である．だから $10^9 + 7$ の倍数でない，すべての整数と互いに素となる．

4.3 二分累乗法

次の手順により p^{20} を求める.

$$p^1\text{を二乗すれば}\,p^2\text{が求まる.}$$
$$p^2\text{を二乗すれば}\,p^4\text{が求まる.}$$
$$p^4\text{を二乗すれば}\,p^8\text{が求まる.}$$
$$p^8\text{を二乗すれば}\,p^{16}\text{が求まる.}$$

ここまでの計算回数は $4 = \log\ 20$ であり，あとは

$$p^{20} = p^{4+16} = p^4 p^{16}$$

を計算する.
実装は

```
1  long long bi_pow(long long p, long long n){
2      int ret=1;
3      while(n>0){
4        if(n & 1){ ret*=p; }
5        p=p*p;
6        n>>=1;
7      }
8      return ret;
9  }
```

5　種々の定理

5.1　中国剰余定理 CRT Chinese remainder theorem

与えられた k 個の整数 $m_1, m_2, \ldots, m_k$ がどの二つも互いに素ならば、任意に与えられる整数 $a_1, a_2, ..., a_k$ に対し

$$\begin{aligned} x \equiv & a_1 \mod m_1 \\ x \equiv & a_2 \mod m_2 \\ & \vdots \\ x \equiv & a_k \mod m_k \end{aligned}$$

を満たす整数 x が $m_1 m_2 \cdots m_k$ を法として一意的に存在する。

この定理で $k = 2$ とした補助定理は以下の通り.
m, n が互いに素のとき

$$\begin{aligned} x \equiv & a \mod n \\ x \equiv & b \mod m \end{aligned}$$

を満たす整数 x が mn を法として一意的に存在する.
補助定理の証明
解の存在．実際に解を構成する．m, n が互いに素であるから，不定方程式

$$mX + nY = 1$$

に整数解 (X, Y) が存在する．このとき

$$x = amX + bnY$$

が，補助定理の 2 式を両方満たす.
一意性. 相異なる解が 2 つあると仮定し，$x = x_1, x_2$ とおく．また，$0 \leq x_1, x_2 < mn$ とする．いま

$$\begin{aligned} x_1 - x_2 \equiv & 0 \pmod n \\ x_1 - x_2 \equiv & 0 \pmod m \\ x_1 - x_2 < & mn \end{aligned}$$

が成り立つ．n と m は互いに素だから $x_1 - x_2$ は mn の 倍数．不等式から,

$$x_1 - x_2 = 0$$

とわかる．しかし，これは相異なる 2 解に矛盾する．よって一意性を示した.

(証明終)

$k = n$ となる場合，数学的帰納法により示される．

5.2 ウィルソンの定理 Wilson's theorem

p を素数として

$$(p-1)! \equiv -1 \pmod{p}$$

5.3 リュカの定理 Lucas's theorem

p を素数として

$${}_mC_n \equiv \prod_{i=0}^{k} {}_{m_i}C_{n_i} \pmod{p}$$

ただし

$$n = \sum_{i=0}^{k} n_i p^i, \quad m = \sum_{i=0}^{k} m_i p^i$$

である．つまり p 進数表記したとき

$$n = n_k n_{k-1} n_{k-2} \dots n_1 n_0$$

$$m = m_k m_{k-1} m_{k-2} \dots m_1 m_0$$

となる．

6　参考 url

全般

https://qiita.com/drken/items/3b4fdf0a78e7a138cd9a

中国剰余定理

https://qiita.com/drken/items/ae02240cd1f8edfc86fd

ウィルソンの定理証明

https://manabitimes.jp/math/744

二項係数

https://drken1215.hatenablog.com/entry/2018/06/08/210000

7　問題

7.1　B - Factorial Yen Coin

高橋王国では 1！円硬貨, 2！円硬貨, …, 10！円硬貨が流通しています。ここで、N！= 1 ×2 ×…× N です。

高橋君は全ての種類の硬貨を 100 枚ずつ持っており、P 円の商品をお釣りが出ないようにちょうどの金額を支払って買おうとしています。

問題の制約下で条件を満たす支払い方は必ず存在することが証明できます。

最小で何枚の硬貨を使えば支払うことができますか？

制約

$$1 \leq P \leq 10$$

P は整数である。

https://atcoder.jp/contests/abc208/tasks/abc208_b

7.2 中華風 (Easy)

3 個の整数の 2 つ組 $(X_1, Y_1)), (X_2, Y_2)), (X_3, Y_3))$ が与えられるので

$$x \equiv X_i \pmod{Y_i}$$

$i = 1, 2, 3$
を全て満たす最小の正整数 x を求める．存在しないなら，-1 を出力．
制約

$$0 \leq X_i < Y_i \leq 10^6$$

`https://yukicoder.me/problems/447`

7.3 - チップ・ストーリー ～黄金編～

1 以上 10^{12} 以下の秘密の整数 N を知りたい．以下のヒントが与えれる．

> $a_2,\ a_3,\ a_4\ \ldots, a_{30}$の 29 個の値
> ただし, a_iは「整数 N を i 進数表示したときの各位の和」である．
> 例えば, $N = 1123$ のとき, N を 4 進数で表すと 101203 となるため,
> $a_4 = 1 + 2 + 1 + 2 + 0 + 3 = 7$ となる．

ヒントをもとにして秘密の整数 N を当てなさい.
制約
$a_2,\ a_3,\ a_4\ \ldots, a_{30}$ は全て 1 以上 500 以下の整数

https://atcoder.jp/contests/ddcc2019-qual/tasks/ddcc2018_qual_d

7.4 J. Ceizenpok ' s formula

$$ {}_nC_k \mod m $$

を求める.
制約

$$ 1 \leq n \leq 10^{18} $$
$$ 0 \leq k \leq n $$
$$ 2 \leq m \leq 1000000 $$

https://codeforces.com/gym/100633/problem/J

7.5 E. Congruence Equation

整数 a,b,p,x が与えられる。

$$Na^N \equiv b \pmod{p}$$

が成立するような 1 以上 x 以下の整数 N を数え上げよ。
制約

$2 \leq p \leq 10^6 + 3$
p は素数
$1 \leq a, b \leq p$
$1 \leq x \leq 10^{12}$

http://codeforces.com/contest/919/problem/E