Počítanie v Z_p a Čínska zvyšková veta

Eduard Batmendijn

29. februára 2016

Abstrakt

Na tejto prednáške si ukážeme, ako počítať v zvyškových triedach modulo nejaké prirodzené číslo. Tiež si ukážeme, ako sa dá efektívne riešiť systém kongruencií.

Poznámka 1. Moduliť budeme vždy kladnými číslami. Modulo zo záporných číslel definujeme tak, aby $x \mod y$ bolo vždy z rozsahu $0, 1, \ldots, y-1$. Operáciu "modulo" budeme značiť znakom %. Množinu $\{0, 1, \ldots, n-1\}$ budeme označovať $\mathbb{Z}_n$.

Počítanie v Z_p

Tvrdenie 1. (triviality):
Majme ľubovoľné kladné celé číslo n. Pre každé dve celé čísla x, y platí:

$$(x + y) \,\%\, n = ((x \,\%\, n) + (y \,\%\, n)) \,\%\, n$$
$$(x - y) \,\%\, n = ((x \,\%\, n) - (y \,\%\, n)) \,\%\, n$$
$$(x \cdot y) \,\%\, n = ((x \,\%\, n) \cdot (y \,\%\, n)) \,\%\, n$$

Úloha 1. (Checkpoint): Dnes je pondelok. Aký deň bude o 5^n dní? (číslo $n \leqslant 10^6$ máte na vstupe).

A čo delenie?

Tvrdenie 2. (o inverzných prvkoch):

Nech n je ľubovoľné kladné celé číslo. Potom pre každé $x \in \mathbb{Z}_n$ nesúdeliteľné s n existuje práve jedno $y \in \mathbb{Z}_n$ také, že

$$x \cdot y \equiv 1 \pmod{n}$$

Toto y sa často značí aj ako x^{-1}.

Veta 1. (Malý Fermat):
Nech p je ľubovoľné prvočíslo a nech a je ľubovoľné celé číslo nedeliteľné p. Potom:

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

Veta 2. (Euler):
Nech n, a sú ľubovoľné nesúdeliteľné kladné celé čísla. Potom:

$$a^{\varphi(n)} \equiv 1 \pmod{n},$$

kde $\varphi(n)$ je počet čísel zo $\mathbb{Z}_n$ nesúdeliteľných s n.

Listing programu (C++)

```
long long power(long long base, long long exp, long long mod);
long long fi(long long mod);

long long sucet(long long a, long long b, long long mod)
{
        return (a + b) % mod;
}

long long rozdiel(long long a, long long b, long long mod)
{
        return (a + mod - b) % mod;
}

long long sucin(long long a, long long b, long long mod)
{
        return (a * b) % mod;
}

long long podiel(long long a, long long b, long long mod)
{
        return (a * power(b, fi(mod)-1, mod)) % mod;
}
```

Rýchle umocňovanie

Úloha 2. (Checkpoint): Dnes je pondelok. Aký deň bude o 5^n dní? (číslo $n \leqslant 10^{18}$ máte na vstupe).

Listing programu (C++)

```
long long power(long long base, long long exp, long long mod)
{
        long long res = 1;
        while(exp > 0)
        {
                if(exp % 2 == 1)
                {
                        res = sucin(res, base, mod);
                }
                exp /= 2;
                base = sucin(base, base, mod);
        }
        return res;
}
```

Čínska zvyšková veta

Veta 3. (Činko Zvyško):

Nech $n_1, n_2, \ldots, n_k$ sú po dvoch nesúdeliteľné kladné celé čísla (ich súčin označme M). Potom pre ľubovoľnú postupnosť celých čísel $a_1, a_2, \ldots, a_k$ existuje práve jedno $x \in \{0, 1, \ldots, M-1\}$ také, že:

$$\begin{aligned} x \equiv & a_1 \pmod{n_1} \\ x \equiv & a_2 \pmod{n_2} \\ & \ldots \\ x \equiv & a_k \pmod{n_k} \end{aligned}$$

Listing programu (C++)

```
struct kongruencia
{
        long long val, mod;
        long long fi;
};

kongruencia spoj_kongruencie(kongruencia s, kongruencia t)
{
        kongruencia res;
        res.mod = s.mod * t.mod;
        res.fi = s.fi * t.fi;
        long long inverz = power(s.mod, t.fi-1, t.mod);
        long long k = sucin(rozdiel(t.val, s.val, res.mod), inverz, res.mod);
        res.val = sucet(sucin(k, s.mod, res.mod),s.val,res.mod);
        return res;
}

kongruencia vyries_sustavu(vector<kongruencia> sustava)
{
        kongruencia res = sustava[0];
        for(int i=1; i < sustava.size(); i++)
        {
                res = spoj_kongruencie(res, sustava[i]);
        }
        return res;
}
```

Hľadanie $\varphi(n)$

Listing programu (C++)

```
long long zrataj_fi(long long n)
{
        long long res = n;
        for(long long p = 2; p * p <= n; p++)
        {
                if(n % p == 0)
                {
                        res /= p;
                        res *= p-1;
                }
                while(n % p == 0)
                {
                        n /= p;
                }
        }
        if(n > 1)
        {
                res /= n;
                res *= n-1;
        }
        return res;
}
```