Talnafræði

Leifareikningur

Bergur Snorrason

17. mars 2021

Leifareikningur

- Látum a og m vera jákvæðar heiltölur.
- Alltaf eru til jákvæðar heiltölur $r < m$ og q þannig $a = q \cdot m + r$.
- Við segjum þá að r sé *leif a með tilliti til m*.
- Við skrifum svo $b = a \mod m$ ef a og b hafa sömu leif með tilliti til m.
- Flest forritunarmál reikna þessa leif með `a%m`.
- Gerum nú ráð fyrir að við séum með jákvæðar heiltölur a_1, a_2, m, og $r_1 = a_1 \mod m$ og $r_2 = a_2 \mod m$.
- Þá gildir að

$$r_1 + r_2 = a_1 + a_2 \mod m$$

og

$$r_1 \cdot r_2 = a_1 \cdot a_2 \mod m.$$

- Þið þurfið að passa ykkur ef þið eruð með neikvæðar tölur.
- Til dæmis er ekki skilgreint hverju `(-10)%3` skilar í `C`.
- Við vitum ekki hvort það skili `-1` eða `2`.
- Til að komast í kringum þessa óvissu notum við frekar `(a%m + m)%m` ef `a` getur verið neikvæð.
- Þetta virkar því `a%m + m` verður alltaf jákvæð.

- Einnig þarf að passa sig að tölurnar verði ekki of stórar.
- Til dæmis er algengt í keppnisforritun að reikna leif með tilliti til $m = 10^9 + 7$.
- Takið eftir að m er ekki of stór fyrir `int`.
- Ef við erum með tvær `int` tölur, `a` og `b`, og viljum reikna `(a*b)%m` þá gæti `a*b` orðið of stór fyrir `int`.
- Til að komast hjá þessu þurfum við að nota `long long`.
- Ef tölurnar eru `long long` í stað `int` þurfum við að nota `__int128`.

```
typedef long long ll;
typedef __int128 lll;

ll bigprod(ll x, ll y, ll m)
{
    return ((lll)x*y)%m;
}
```

- Stundum þurfum við að geta deilt í leifareikningi.
- Þetta er ekki hægt að gera með hefðbundinni deilingu.
- Við látum b^{-1} tákna þá tölu sem uppfyllir að $1 = b \cdot b^{-1} \mod m$.
- Þessi tala er ekki alltaf til.
- Hún er þó alltaf til ef m er frumtala.
- Við köllum b^{-1} *margföldunar andhverfu b með tilliti til m*.
- Við skrifum svo stundum a/b í stað ab^{-1}.
- En hvernig finnum við þessa tölu?

- Látum p vera frumtölu.
- Litla setning Fermats segir okkur að $a^p = a \mod p$.
- Ef við margföldum báðum megin með a^{-2} fæst að $a^{p-2} = a^{-1} \mod p$.
- Svo eina sem við þurfum að gera er að reikna $a^{p-2} \mod p$.
- Gerum ráð fyrir að við séum með fall `modpow(x, n, m)` sem reiknar $x^n \mod m$ (við útfærum það á eftir).

```
ll mulinv(ll a, ll p)
{
    return modpow(a, p - 2, p);
}
```

- Tímaflækjan á þessari aðferð verður síðan sú sama og tímaflækjan á `modpow(...)`.

- Til að finna $a^{-1} \mod m$ ef m er ekki frumtala er ögn flóknara.
- Við skoðum það á eftir þegar við skoðum reiknirit Evklíðs.