C++ Snippets

Muhammad Samir Assawalhy

August 16, 2024

Contents

1 Math Miscellaneous

1.1 Master's theorem

Consider the recurrence relation:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where $a \geq 1$ and $b > 1$. The solution to this recurrence relation depends on the asymptotic behavior of $f(n)$ compared to $n^{\log_b a}$:

1. If $f(n) = O(n^c)$ where $c < \log_b a$, then:
$$T(n) = O(n^{\log_b a})$$
2. If $f(n) = \Theta(n^{\log_b a})$, then:
$$T(n) = \Theta(n^{\log_b a} \log n)$$
3. If $f(n) = \Omega(n^c)$ where $c > \log_b a$, and if $af\left(\frac{n}{b}\right) \leq kf(n)$ for some constant $k < 1$ and sufficiently large n, then:
$$T(n) = \Theta(f(n))$$

1.2 Euler Totient $\phi(n)$

- $\phi(n)$ is the count of co-primes of n from 1 to n.
- So that $\phi(4) = 2$ which are 3 and 1.
- $\phi(p) = p - 1$ for a prime p.

The following is Euler's theorem which can be used to compute modular inverse of a only if $\gcd(a, m) = 1$.

$$a^{\phi(m)} \equiv 1 \mod m$$

When $g = \gcd(a, m) > 1$ and you want to compute a^x when x is too large to do with binary exponentiation:

$$\begin{aligned}
a^{\phi(m)} &\not\equiv 1 \pmod{m} \\
a^x &\equiv \left(\frac{a}{g}\right)^x \cdot g^x \pmod{m} \\
a^{\phi\left(\frac{m}{g}\right)} &\equiv g^{\phi\left(\frac{m}{g}\right)} \equiv 1 \pmod{\frac{m}{g}} \\
g \cdot g^{\phi\left(\frac{m}{g}\right)} &\equiv g \pmod{m} \\
a^x &\equiv \left(\frac{a}{g}\right)^{x \bmod m} \cdot g^x \pmod{m}
\end{aligned}$$

1.3 Modular arithmetics

Get inverse of number module prime p for all values in $[1, x]$:

1: Initialize inv array with 1s of size $x + 1$
2: **for** $i \leftarrow 2$ to x **do**
3: $\quad inv_i \leftarrow -\frac{p}{i} \cdot inv_{p \bmod i} \mod p$