Fundamental formulas, theorems and snippets

Oscar Burga

Dec 2019

1 Modular Arithmetic

1.1 General formulas

- $ab \bmod ac = a(b \bmod c)$

1.2 Fermat's little theorem

For any integer a and prime number p:

- $a^p \equiv a \pmod{p} \quad \Rightarrow \quad (a^p - a)$ is an integer multiple of p
- $a^{p-1} \equiv 1 \pmod{p} \quad \Rightarrow \quad (a^{p-1} - 1)$ is an integer multiple of p

1.2.1 Carmichael numbers

A Carmichael number is an **odd** composite number n such that:

- $b^{n-1} \equiv 1 (\mathrm{mod} n)$ for all integers $b \in [2, n-1]$ such that b is coprime to n
- A positive, composite integer n is a Carmichael number if and only if n is square-free (no perfect square divides n) and for all prime factors p of n, $p - 1$ divides $n - 1$.
- There are no Carmichael numbers with exactly two prime divisors.

1.3 Euler's theorem: A generalization of Fermat's little theorem

- Euler's totient function ($\phi(n)$): Counts the number of integers $x \in [1, n]$ which are coprime to n.
- Phi-function properties (follows from Fermat's little theorem):
  - $\phi(p) = p - 1$ for prime p
  - $\phi(p^k) = p^k - p^{k-1}$ for prime p
  - $\phi(ab) = \phi(a) \cdot \phi(b)$ for coprime a and b
  - $\phi(ab) = \phi(a) \cdot \phi(b) \cdot \frac{d}{\phi(d)}$ for not coprime a and b, where $d = gcd(a, b)$
  - Number of valid x such that $x \in [1, n)$ and $gcd(x, n) = k$ can be solved as $\phi(\frac{n}{k})$
- Application in Euler's Theorem:
  - $x \equiv y \pmod{\phi(n)} \Rightarrow a^x \equiv a^y \pmod{n}$ for coprime a and n
  - $x^{\phi(m)} \equiv 1 \pmod{m}$ for coprime x and m
  - $x^n \equiv x^{n \bmod \phi(m)} \pmod{m}$ for coprime x and m
  - $x^n \equiv x^{\phi(m) + [n \bmod \phi(m)]} (\mathrm{mod}\ m)$ for arbitrary x, m and $n \geq \log_2 m$

2 Combinatorics

- Permutation:

$$_nP_r = \binom{n}{r} \cdot r! = \frac{n!}{(n-r)!}$$

- Combination:

$$_nC_r = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

$$\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}$$

$$\binom{n}{r} = \sum_{i=0}^{r} \binom{n-1}{i}$$

- Snippet for Pascal's Triangle (overflow past $\binom{62}{31}$)

```
ll dp[maxn][maxn];
ll comb(ll n, ll r){
  if (r == 0 || n == r) return 1;
  if (dp[n][r] != -1) return dp[n][r];
  dp[n][r] = comb(n-1, r-1) + comb(n-1, r);
  return dp[n][r];
}
```