Team Notebook

November 29, 2024

Contents

1 !!! Kata-Kata Bijak !!!

- Izin sumbit ya
- jangan ngerjain sorang-sorang
- Jangan lupa solat
- Wajarlah manusia bukan nabi boy
- Jangan takut gambling - hash dan random adalah teman
- Soal Math - Ingat kata guru SMA, cari pola, atau bikin rumus keajaiban (siap king), atau bikin brute force
- Kalau N ¡= 100 ada kemungkinan max-flow
- KALO HASHING MINIMAL DOUBLE HASH!
- Jangan stres – have fun aja
- Kalau ragu, verify ide ke teman-teman
- OP TI MIS dan SE MA NGAT

2 Pengingat

2.1 combinatorics

mathtools

- $\sum_{k=0}^{n} k^2 = n(n+1)(2n+1)/6$
- $\sum_{k=0}^{n} k^3 = n^2(n+1)^2/4$
- $\sum_{k=0}^{n} k^4 = (6n^5 + 15n^4 + 10n^3 - n)/30$
- $\sum_{k=0}^{n} k^5 = (2n^6 + 6n^5 + 5n^4 - n^2)/12$
- $\sum_{k=0}^{n} kx^k = (x - (n+1)x^{n+1} + nx^{n+2})/(x-1)^2$
- $(n+1)^{k+1} - 1 = \sum_{m=1}^{n}((m+1)^{k+1} - m^{k+1}) = \sum_{p=0}^{k}\binom{k+1}{p}(1^p + 2^p + ... + n^p)$
- $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}$
- $\sum_{k=0}^{n} k^2\binom{n}{k} = (n+n^2)2^{n-2}$
- $\sum_{j=0}^{k}\binom{m}{j}\binom{n-m}{k-j} = \binom{n}{k}$
- $\sum_{m=0}^{n}\binom{m}{k} = \binom{n+1}{k+1}$
- $\sum_{j=0}^{m}\binom{m}{j}^2 = \binom{2m}{m}$
- $\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor}\binom{n-k}{k} = F(n+1)$
- $\sum_{i=0}^{n} i\binom{n}{i}^2 = \frac{n}{2}\binom{2n}{n}$
- $\sum_{i=0}^{n} i^2\binom{n}{i}^2 = n^2\binom{2n-2}{n-1}$
- $\sum_{k=q}^{n}\binom{n}{k}\binom{k}{q} = 2^{n-q}\binom{n}{q}$
- $\sum_{k=-a}^{a}(-1)^k\binom{2a}{k+a}^3 = \frac{(3a)!}{(a!)^3}$
- $\sum_{k=-a}^{a}(-1)^k\binom{a+b}{a+k}\binom{b+c}{b+k}\binom{c+a}{c+k} = \frac{(a+b+c)!}{a!b!c!}$

2.2 theorem

Cayley's Formula

There are n^{n-2} spanning trees of a complete graph with n labeled vertices.

Derangement

A permutation of the elements of a set such that none of the elements appear in their original positions. $F(n) = (n-1) * (F(n-1) + F(n-2))$. $F(0) = 1$. $F(1) = 0$.

Euler's Formula for Planar Graph

$V - E + F = 2$, where V = vertices, E = edges, F = faces

Pick's Theorem

$A = i + b/2 - 1$, where A = area, i = internal points, b = border points

Spanning Tree of Complete Bipartite Graph

$N^{M-1} * M^{N-1}$, where N = row and M = column

Pythagorean Triples

Integer solutions of $x^2 + y^2 = z^2$. All relatively prime triples are given by: $x = 2mn, y = m^2 - n^2, z = m^2 + n^2$, where $m > n$, $gcd(m,n) = 1$, and $m! = n \pmod 2$.

Moser's Circle

Determine the number of pieces into which a circle is divided if n points on its circumference are joined by chords with no three internally concurrent. Solution: $g(n) = nC4 + nC2 + 1$

Kirchoff Matrix Theorem

Let matrix $T = [t_{ij}]$, where t_{ij} is the number of multi-edges between i and j, for $i \neq j$, and $t_{ii} = -deg[i]$. Number of spanning trees of a graph is equal to the determinant of a matrix obtained by deleting any k-th row and column from T.

Euler's Theorem

$a^{phi(n)} = 1 \pmod n$, if $gcd(a,n) = 1$.

Wilson's Theorem

p is prime iff $(p-1) \neq -1 \pmod p$.

Pisano Period

Periodicity of fibonacci modulo m.

- $pi(p^k) = p^{k-1} * pi(p)$
- $pi(2) = 3, pi(5) = 20$
- if $p \equiv 1$ or $p \equiv 9$ in modulo 10, $pi(p)$ divides $p-1$
- if $p \equiv 3$ or $p \equiv 7$ in modulo 10, $pi(p)$ divides $2p-1$
- $pi(a*b) = lcm(pi(a), pi(b))$ if $gcd(a,b) = 1$

Misere Nim

Nim where the winner is the one who can't move. In a nim game with piles $(n_1, n_2, \cdots, n_k)$, **second** player wins iff some $n_i > 1$ and $(n_1 \oplus n_2 \oplus ... \oplus n_k) = 0$ or all $n_i \leq 1$ and $n_1 \oplus n_2 \oplus ... \oplus n_k = 1$.

3 Template

3.1 Dinic

```cpp
struct FlowEdge {
    int v, u;
    long long cap, flow = 0;
    FlowEdge(int v, int u, long long cap) : v(v), u(u), cap(cap) {}
};

struct Dinic {
    const long long flow_inf = 1e18;
    vector<FlowEdge> edges;
    vector<vector<int>> adj;
    int n, m = 0;
    int s, t;
    vector<int> level, ptr;
    queue<int> q;

    Dinic(int n, int s, int t) : n(n), s(s), t(t) {
        adj.resize(n);
        level.resize(n);
        ptr.resize(n);
    }

    void add_edge(int v, int u, long long cap) {
        edges.emplace_back(v, u, cap);
        edges.emplace_back(u, v, 0);
        adj[v].push_back(m);
        adj[u].push_back(m + 1);
        m += 2;
    }

    bool bfs() {
        while (!q.empty()) {
            int v = q.front();
            q.pop();
            for (int id : adj[v]) {
                if (edges[id].cap - edges[id].flow < 1)
                    continue;
                if (level[edges[id].u] != -1)
                    continue;
                level[edges[id].u] = level[v] + 1;
                q.push(edges[id].u);
            }
        }
        return level[t] != -1;
    }

    long long dfs(int v, long long pushed) {
        if (pushed == 0)
            return 0;
        if (v == t)
            return pushed;
        for (int& cid = ptr[v]; cid < (int)adj[v].size(); cid++) {
            int id = adj[v][cid];
            int u = edges[id].u;
            if (level[v] + 1 != level[u] || edges[id].cap - edges[id].flow < 1)
                continue;
            long long tr = dfs(u, min(pushed, edges[id].cap - edges[id].flow));
            if (tr == 0)
                continue;
            edges[id].flow += tr;
            edges[id ^ 1].flow -= tr;
            return tr;
        }
        return 0;
    }

    long long flow() {
        long long f = 0;
        while (true) {
            fill(level.begin(), level.end(), -1);
            level[s] = 0;
            q.push(s);
            if (!bfs())
                break;
            fill(ptr.begin(), ptr.end(), 0);
            while (long long pushed = dfs(s, flow_inf)) {
                f += pushed;
            }
        }
        return f;
    }
};
```