Team Contest Reference
Team: stoptryharding

Moritz Hoffmann
Ian Pösse
Marcel Wienöbst

Contents

1 ds **1**

2 graph **1**

3 math **1**
3.1 geometry lib 1

4 misc **2**

5 Math **2**
5.1 Tree 2
5.2 Divisability Explanation 2
5.3 Combinatorics 2
5.4 Polynomial Interpolation 2
5.4.1 Theory 2
5.5 Fibonacci Sequence 2
5.5.1 Binet's formula 2
5.5.2 Generalization 2
5.5.3 Pisano Period 2
5.6 Reihen 2
5.7 Binomialkoeffizienten 2
5.8 Catalanzahlen 2
5.9 Geometrie 2
5.10 Zahlentheorie 3
5.11 Faltung 3

1 ds

2 graph

3 math

3.1 geometry lib

```
// this library has been copied from https://github.com/SuprDewd/T-414-AFLV
#include <complex>
using namespace std;
#define P(p) const point &p
#define L(p0, p1) P(p0), P(p1)
#define C(p0, r) P(p0), double r
#define PP(pp) pair<point,point> &pp
typedef complex<double> point;
const double pi = acos(-1.0);
const double EPS = 1e-9;
double dot(P(a), P(b)) {
    return real(conj(a) * b);
}
double cross(P(a), P(b)) {
    return imag(conj(a) * b);
}
point rotate(P(p), double radians = pi / 2, P(about) = point(0,0)) {
    return (p - about) * exp(point(0, radians)) + about;
}
point proj(P(u), P(v)) {
    return dot(u, v) / dot(u, u) * u;
}
point normalize(P(p), double k = 1.0) {
    return abs(p) == 0 ? point(0,0) : p / abs(p) * k;
}
bool parallel(L(a, b), L(p, q)) {
    return abs(cross(b - a, q - p)) < EPS;
}
double ccw(P(a), P(b), P(c)) {
    return cross(b - a, c - b);
}
bool collinear(P(a), P(b), P(c)) { return abs(ccw(a, b, c)) < EPS; }
double angle(P(a), P(b), P(c)) {
    return acos(dot(b - a, c - b) / abs(b - a) / abs(c - b));
}
bool intersect(L(a, b), L(p, q), point &res, bool segment = false) {
```

```
    // NOTE: check for parallel/collinear lines before calling this function
    point r = b - a, s = q - p;
    double c = cross(r, s), t = cross(p - a, s) / c, u = cross(p - a, r) / c;
    if (segment && (t < 0-EPS || t > 1+EPS || u < 0-EPS || u > 1+EPS))
        return false;
    res = a + t * r;
    return true;
}
point closest_point(L(a, b), P(c), bool segment = false) {
    if (segment) {
        if (dot(b - a, c - b) > 0) return b;
        if (dot(a - b, c - a) > 0) return a;
    }
    double t = dot(c - a, b - a) / norm(b - a);
    return a + t * (b - a);
}

typedef vector<point> polygon;
#define MAXN 1000
point hull[MAXN];
bool cmp(const point &a, const point &b) {
    return abs(real(a) - real(b)) > EPS ?
        real(a) < real(b) : imag(a) < imag(b); }
int convex_hull(vector<point> p) {
    int n = p.size(), l = 0;
    sort(p.begin(), p.end(), cmp);
    for (int i = 0; i < n; i++) {
        if (i > 0 && p[i] == p[i - 1])
            continue;
        while (l >= 2 && ccw(hull[l - 2], hull[l - 1], p[i]) >= 0)
            l--;
        hull[l++] = p[i];
    }
    int r = l;
    for (int i = n - 2; i >= 0; i--) {
        if (p[i] == p[i + 1])
            continue;
        while (r - l >= 1 && ccw(hull[r - 2], hull[r - 1], p[i]) >= 0)
            r--;
        hull[r++] = p[i];
    }
    return l == 1 ? 1 : r - 1;
}
```

MD5: `3563f20cd2010aee48a137414d73506c` | $\mathcal{O}(?)$

4 misc

5 Math

5.1 Tree

Diameter: BFS from any node, then BFS from last visited node. Max dist is then the diameter. Center: Middle vertex in second step from above.

5.2 Divisability Explanation

$D \mid M \Leftrightarrow D \mid$ `digit_sum(M, k, alt)`, refer to table for values of D, k, alt.

5.3 Combinatorics

- Variations (ordered): k out of n objects (permutations for $k = n$)
  - without repetition: $M = \{(x_1, \ldots, x_k) : 1 \leq x_i \leq n,\ x_i \neq x_j \text{ if } i \neq j\}$, $|M| = \frac{n!}{(n-k)!}$
  - with repetition: $M = \{(x_1, \ldots, x_k) : 1 \leq x_i \leq n\}, |M| = n^k$
- Combinations (unordered): k out of n objects
  - without repetition: $M = \{(x_1, \ldots, x_n) : x_i \in \{0,1\},\ x_1 + \ldots + x_n = k\}, |M| = \binom{n}{k}$
  - with repetition: $M = \{(x_1, \ldots, x_n) : x_i \in \{0,1,\ldots,k\},\ x_1 + \ldots + x_n = k\}, |M| = \binom{n+k-1}{k}$
- Ordered partition of numbers: $x_1 + \ldots + x_k = n$ (i.e. 1+3 = 3+1 = 4 are counted as 2 solutions)
  - #Solutions for $x_i \in \mathbb{N}_0$: $\binom{n+k-1}{k-1}$
  - #Solutions for $x_i \in \mathbb{N}$: $\binom{n-1}{k-1}$
- Unordered partition of numbers: $x_1 + \ldots + x_k = n$ (i.e. 1+3 = 3+1 = 4 are counted as 1 solution)
  - #Solutions for $x_i \in \mathbb{N}$: $P_{n,k} = P_{n-k,k} + P_{n-1,k-1}$ where $P_{n,1} = P_{n,n} = 1$
- Derangements (permutations without fixed points): $!n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!} = \lfloor \frac{n!}{e} + \frac{1}{2} \rfloor$

5.4 Polynomial Interpolation

5.4.1 Theory

Problem: for $\{(x_0, y_0), \ldots, (x_n, y_n)\}$ find $p \in \Pi_n$ with $p(x_i) = y_i$ for all $i = 0, \ldots, n$.
Solution: $p(x) = \sum\limits_{i=0}^{n} \gamma_{0,i} \prod\limits_{j=0}^{i-1} (x - x_i)$ where $\gamma_{j,k} = y_j$ for $k = 0$ and $\gamma_{j,k} = \frac{\gamma_{j+1,k-1} - \gamma_{j,k-1}}{x_{j+k} - x_j}$ otherwise.
Efficient evaluation of $p(x)$: $b_n = \gamma_{0,n}$, $b_i = b_{i+1}(x - x_i) + \gamma_{0,i}$ for $i = n-1, \ldots, 0$ with $b_0 = p(x)$.

5.5 Fibonacci Sequence

5.5.1 Binet's formula

$\begin{pmatrix} f_n \\ f_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow f_n = \frac{1}{\sqrt{5}}(\phi^n - \tilde{\phi}^n)$ where $\phi = \frac{1+\sqrt{5}}{2}$ and $\tilde{\phi} = \frac{1-\sqrt{5}}{2}$.

5.5.2 Generalization

$g_n = \frac{1}{\sqrt{5}}(g_0(\phi^{n-1} - \tilde{\phi}^{n-1}) + g_1(\phi^n - \tilde{\phi}^n)) = g_0 f_{n-1} + g_1 f_n$ for all $g_0, g_1 \in \mathbb{N}_0$

5.5.3 Pisano Period

Both $(f_n \mod k)_{n\in\mathbb{N}_0}$ and $(g_n \mod k)_{n\in\mathbb{N}_0}$ are periodic.

5.6 Reihen

$\sum\limits_{i=1}^{n} i = \frac{n(n+1)}{2}, \sum\limits_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \sum\limits_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$
$\sum\limits_{i=0}^{n} c^i = \frac{c^{n+1}-1}{c-1}, c \neq 1, \sum\limits_{i=0}^{\infty} c^i = \frac{1}{1-c}, \sum\limits_{i=1}^{n} c^i = \frac{c}{1-c}, |c| < 1$
$\sum\limits_{i=0}^{n} ic^i = \frac{nc^{n+2}-(n+1)c^{n+1}+c}{(c-1)^2}, c \neq 1, \sum\limits_{i=0}^{\infty} ic^i = \frac{c}{(1-c)^2}, |c| < 1$

5.7 Binomialkoeffizienten

$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$, $\binom{n}{m}\binom{m}{k} = \binom{n}{k}\binom{n-k}{m-k}$, $\binom{m+n}{r} = \sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k}$ and in general, $n_1 + \cdots + n_p = \sum_{k_1+\cdots+k_p=m}\binom{n_1}{k_1}\cdots\binom{n_p}{k_p}$

5.8 Catalanzahlen

$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!n!}$
$C_0 = 1, C_{n+1} = \sum\limits_{k=0}^{n} C_k C_{n-k}, C_{n+1} = \frac{4n+2}{n+2} C_n$

5.9 Geometrie

Polygonfläche: $A = \frac{1}{2}(x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + \cdots + x_{n-1}y_n - x_ny_{n-1} + x_ny_1 - x_1y_n)$

5.10 Zahlentheorie

Chinese Remainder Theorem: Es existiert eine Zahl C, sodass: $C \equiv a_1 \mod n_1, \cdots, C \equiv a_k \mod n_k, \text{ggt}(n_i, n_j) = 1, i \neq j$
Fall $k = 2$: $m_1n_1 + m_2n_2 = 1$ mit EEA finden.
Lösung ist $x = a_1m_2n_2 + a_2m_1n_1$.
Allgemeiner Fall: iterative Anwendung von $k = 2$
Eulersche φ-Funktion: $\varphi(n) = n\prod_{p|n}(1 - \frac{1}{p}), p$ prim
$\varphi(p) = p - 1, \varphi(pq) = \varphi(p)\varphi(q), p, q$ prim
$\varphi(p^k) = p^k - p^{k-1}, p, q$ prim, $k \geq 1$
Eulers Theorem: $a^{\varphi(n)} \equiv 1 \mod n$
Fermats Theorem: $a^p \equiv a \mod p, p$ prim

5.11 Faltung

$(f * g)(n) = \sum\limits_{m=-\infty}^{\infty} f(m)g(n-m) = \sum\limits_{m=-\infty}^{\infty} f(n-m)g(m)$