Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0503 – Шабанов Тимофей Константинович

A	B	C	D	E	F	sum
30	100	100	75	20	91	416

Task A

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

void solve() {
    int a, b, n;
    cin >> a >> b >> n;

    ll ans = 0;
    for (int mask = 1; mask < (1 << n); mask++) {
        vector<int> pos;
        for (int i = 0; i < n; i++) if (mask >> i & 1) pos.push_back(i+1)
    ;
        bool ok = true;
        for (int i = 0; i < pos.size(); i++) {
            int x;
            if (i + 1 == pos.size()) {
                x = (n - pos[i]) + pos[0];
            } else {
                x = pos[i+1] - pos[i];
            }
            ok &= x == a || x == b;
        }
        ans += ok;
    }
```

```
    cout << ans << '\n';
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task B

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

void solve() {
    int n;
    cin >> n;

    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;

    int x2, y2, z2;
    cin >> x2 >> y2 >> z2;

    auto dist = [&](int x1, int y1, int z1, int x2, int y2, int z2) {
        int res = 1e9;
        res = min(res, abs(x2 - x1) + abs(y2 - y1));
        res = min(res, abs(x2 - x1) + abs(z2 - z1));
        res = min(res, abs(y2 - y1) + abs(z2 - z1));
        return res;
    };

    int ans = dist(x1, y1, z1, x2, y2, z2);

    // x = -n
    for (int y = 0; y <= n; y++) {
        int x = -n;
        int z = -(x + y);
        int rx = n, ry = y - n, rz = z - n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    // x = n
```

```
    for (int y = -n; y <= 0; y++) {
        int x = n;
        int z = -(x + y);
        int rx = -n, ry = y + n, rz = z + n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    // y = -n
    for (int x = 0; x <= n; x++) {
        int y = -n;
        int z = -(x + y);
        int rx = x - n, ry = n, rz = z - n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    // y = n
    for (int x = -n; x <= 0; x++) {
        int y = n;
        int z = -(x + y);
        int rx = x + n, ry = -n, rz = z + n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    // z = -n
    for (int x = 0; x <= n; x++) {
        int z = -n;
        int y = -(x + z);
        int rx = x - n, ry = y - n, rz = n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    // z = n
    for (int x = -n; x <= 0; x++) {
        int z = n;
        int y = -(x + z);
        int rx = x + n, ry = y + n, rz = -n;
        ans = min(ans, dist(x1, y1, z1, x, y, z) + dist(rx, ry, rz, x2,
    y2, z2));
    }

    cout << ans << '\n';
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();
```

```
    return 0;
}
```

Task C

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

constexpr ll MOD = (ll)1e18+9;
constexpr int P = 239017;

ll add(ll a, ll b) {
    if ((a += b) >= MOD) {
        a -= MOD;
    }
    return a;
}
ll sub(ll a, ll b) {
    if ((a -= b) < 0) {
        a += MOD;
    }
    return a;
}
ll mul(ll a, ll b) {
    return (__int128)a * b % MOD;
}

string bin(ll x) {
    string res;
    while (x > 0) {
        res += (x % 2) + '0';
        x /= 2;
    }
    return res;
}

void solve() {
    string o;
    cin >> o;

    if (o == "first") {
        int n;
        cin >> n;

        string s;
        cin >> s;
```

```
        ll h = 0;
        for (auto c : s) h = add(mul(h, P), c);
        cout << bin(h) << '\n';
    } else {
        int n;
        cin >> n;

        string t;
        cin >> t;

        string hint;
        cin >> hint;

        vector<ll> pre(n+1);
        for (int i = 0; i < n; i++) {
            pre[i+1] = add(mul(pre[i], P), t[i]);
        }
        constexpr int N = 1e5+7;
        vector<ll> pw(N+1);
        pw[0] = 1;
        for (int i = 0; i < N; i++) {
            pw[i+1] = mul(pw[i], P);
        }

        auto hash = [&](int l, int r) {
            if (l > r) return 0ll;
            return sub(pre[r], mul(pre[l], pw[r-l]));
        };

        ll need = 0;
        for (int i = 0; i < hint.size(); i++) {
            if (hint[i] == '1') need += 1ll << i;
        }

        for (int k = 0; k < n; k++) {
            if (add(mul(hash(k, n), pw[k]), hash(0, k)) == need) {
                cout << "YES\n";
                return;
            }
        }
        cout << "NO\n";
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task D

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

// vector<int> p = {4, 3, 2, 1};

void solve() {
    int n;
    cin >> n;

    vector<int> pos(n+1); // position of p[i]
    iota(pos.begin(), pos.end(), 0);

    auto ask = [&](int i, int j) {
        int x = -1, y = -1;
        for (int k = 1; k <= n; k++) {
            if (pos[k] == i) x = k;
            if (pos[k] == j) y = k;
        }
        swap(pos[x], pos[y]);
        for (int k = 1; k <= n; k++) {
            pos[k]--;
            if (pos[k] == 0) {
                pos[k] = n;
            }
        }

        // vector<int> res(n+1);
        // for (int i = 1; i <= n; i++) {
        //     res[pos[i]] = P[i];
        // }
        // cout << "res: ";
        // for (auto x : res) cout << x << ' ';
        // cout << endl;

        cout << i << ' ' << j << endl;
        // {
        //     auto prv = p;
        //     swap(p[i-1], p[j-1]);
        //     rotate(p.begin(), p.begin()+1, p.end());
        //     // cout << "p: ";
        //     // for (auto x : p) cout << x << ' ';
        //     // cout << endl;
        //     char c;
        //     if (p > prv) c = '>';
        //     else if (p == prv) c = '=';
        //     else c = '<';
```

```
    //      return c == '>';
    // }
    {
        // swap(p[i-1], p[j-1]);
        char c;
        cin >> c;
        if (c == '!') exit(0);
        return c == '>';
    }
};

vector<int> inv(n+1);
for (int i = 1; i <= n; i++) {
    for (int j = 1; j < i; j++) {
        for (int k = 1; k < j; k++) {
            ask(2, pos[j]);
            inv[i] += ask(2, pos[k]);
        }
        for (int k = i; k <= n; k++) {
            ask(2, pos[j]);
            inv[i] += ask(2, pos[k]);
        }
    }
    for (int j = i; j <= n; j++) {
        for (int k = i; k < j; k++) {
            ask(2, pos[j]);
            inv[i] += ask(2, pos[k]);
        }
    }
}

// cout << "inv: ";
// for (int i = 1; i <= n; i++) cout << inv[i] << ' ';
// cout << '\n';

vector<int> cur(n+1), q(n+1);
vector<bool> vis(n+1);
for (int i = 1; i < n; i++) {
    int pi = ((n + 1) - (inv[i+1] - inv[i])) / 2;
    q[i] = pi;
    vis[pi] = true;
}
for (int i = 1; i <= n; i++) {
    if (vis[i]) continue;
    for (int j = 1; j <= n; j++) {
        if (q[j] == 0) {
            q[j] = i;
            vis[i] = true;
            break;
        }
    }
}
// cout << '\n';
```

```
    // for (int i = 1; i <= n; i++) cout << cur[i] << ' ';

    // cout << "q: ";
    // for (int i = 1; i <= n; i++) cout << q[i] << ' ';
    // cout << endl;

    vector<int> ord(n);
    iota(ord.begin(), ord.end(), 1);
    sort(ord.begin(), ord.end(), [&](int i, int j) {
        return q[i] > q[j];
    });

    for (auto i : ord) {
        ask(1, pos[i]);
    }
    // cout << "output:";
    // for (auto x : p) cout << x << ' ';
    // cout << endl;
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task E

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

void solve() {
    int n, q;
    cin >> n >> q;

    vector<vector<int>> adj(n);
    for (int i = 0; i < n; i++) {
        int k;
        cin >> k;
        for (int j = 0; j < k; j++) {
            int x;
            cin >> x;
            x--;
            adj[i].push_back(x);
```

```
        }
    }

    vector<int> qry(q);
    for (auto &i : qry) cin >> i;

    int x = 0, T = 0, top = 0;
    while (top < q) {
        while (T < qry[top]) {
            int to = adj[x][0];
            rotate(adj[x].begin(), adj[x].begin()+1, adj[x].end());
            x = to;
            T++;
        }
        cout << x+1 << ' ';
        top++;
    }

    // rotate(a.begin(), a.begin()+1, a.end());
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task F

```
#include <bits/stdc++.h>

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

using namespace std;
using ll = long long;

mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    int h, w;
    cin >> h >> w;

    vector a(h, vector<int>(w));
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            char c;
            cin >> c;
            a[i][j] = c - '0';
```

```
    }
}

ll prod = -1;
vector<string> answer(h, string(w, '.'));

auto relax = [&](vector<string> &table) {
    ll cur = 0;
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            if (table[i][j] == 'L') {
                cur += a[i][j]*a[i][(j+1)%w];
            }
            if (table[i][j] == 'T') {
                cur += a[i][j]*a[(i+1)%h][j];
            }
        }
    }
    if (cur > prod) {
        prod = cur;
        answer = table;
    }
};

{
    vector<string> table(h, string(w, '.'));
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j += 2) {
            table[i][j] = 'L';
            table[i][j+1] = 'R';
        }
    }
    relax(table);
}

{
    vector<string> table(h, string(w, '.'));
    for (int j = 0; j < w; j++) {
        for (int i = 0; i < h; i += 2) {
            table[i][j] = 'T';
            table[i+1][j] = 'B';
        }
    }
    relax(table);
}

{
    for (int rep = 0; rep < 20000; rep++) {
        vector<string> table(h, string(w, '.'));
        for (int i = 0; i < h; i += 2) {
            for (int j = 0; j < w; j += 2) {
                if (rng() % 2) {
                    table[i][j] = table[i+1][j] = 'L';
```

```
                    table[i][j+1] = table[i+1][j+1] = 'R';
                } else {
                    table[i][j] = table[i][j+1] = 'T';
                    table[i+1][j] = table[i+1][j+1] = 'B';
                }
            }
        }
        relax(table);
    }
}

{
    vector<string> table(h, string(w, '.'));
    for (int k = 0; k < h*w/2; k++) {
        int bi = -1, bj = -1, dir = -1, res = -1;
        for (int i = 0; i < h; i++) {
            for (int j = 0; j < w; j++) {
                if (table[i][j] == '.' && table[i][(j+1)%w] == '.') {
                    if (res < a[i][j]*a[i][(j+1)%w]) {
                        bi = i;
                        bj = j;
                        dir = 0;
                        res = a[i][j]*a[i][(j+1)%w];
                    }
                    // table[i][j] = 'L';
                    // table[i][(j+1)%w] = 'R';
                }
                if (table[i][j] == '.' && table[(i+1)%h][j] == '.') {
                    if (res < a[i][j]*a[(i+1)%h][j]) {
                        bi = i;
                        bj = j;
                        dir = 1;
                        res = a[i][j]*a[(i+1)%h][j];
                    }
                    // table[i][j] = 'T';
                    // table[(i+1)%h][j] = 'B';
                }
            }
        }
        if (bi != -1) {
            if (dir == 0) {
                table[bi][bj] = 'L';
                table[bi][(bj+1)%w] = 'R';
            } else {
                table[bi][bj] = 'T';
                table[(bi+1)%h][bj] = 'B';
            }
        }
    }
    relax(table);
}

cout << prod << '\n';
```

```
    for (auto &i : answer) cout << i << '\n';
}

int main() {
    cin.tie(0)->sync_with_stdio(0);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```