Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0527 – Туревич Артём Александрович

A	B	C	D	E	F	sum
100	100	100	100	40	91	531

Task A

```
C = [[0 for _ in range(100)] for __ in range(100)]
for i in range(100):
    C[0][i] = 1
    C[i][i] = 1
for n in range(1, 100):
    for k in range(1, n):
        C[k][n] = C[k][n - 1] + C[k - 1][n - 1]


a, b, n = map(int, input().split())

ans = 0
# n - a
for i in range(n):
    for j in range(n):
        if a * i + b * j == n - a:
            ans += C[i][i + j] * a
# n - b
for i in range(n):
    for j in range(n):
        if a * i + b * j == n - b:
            ans += C[i][i + j] * b
print(ans)
```

Task B

```
#include <bits/stdc++.h>
// tiom4eg's precompiler options
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
```

```cpp
// IO settings
#define fastIO ios_base::sync_with_stdio(false); cin.tie(0)
// Quick types
#define ll long long
#define ld long double
#define ull unsigned long long
#define pii pair <int, int>
#define vi vector <int>
#define mi vector <vector <int>>
// Quick functions
#define endl "\n"
#define F first
#define S second
#define all(a) a.begin(), a.end()
#define sz(a) (int)(a.size())
#define pb push_back
#define mp make_pair
// Quick fors
#define FOR(i, a, b) for (int i = a; i < b; ++i)
#define FORS(i, a, b, c) for (int i = a; i < b; i += c)
#define RFOR(i, a, b) for (int i = a; i >= b; --i)
#define EACH(e, a) for (auto& e : a)
// Pragmas
#ifndef TIOM4EG
#pragma GCC optimize("O3") // let the chaos begin!
#pragma GCC target("avx,avx2,tune=native")
#pragma GCC comment(linker, "/stack:200000000")
#endif
// PBDS
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#define pbds tree <int, null_type, less <int>, rb_tree_tag,
    tree_order_statistics_node_update>
using namespace __gnu_pbds;
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
using namespace std;
mt19937 rng(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
#define int long long
constexpr int INF = 1e9 + 7, MD = 998244353, MAX = 400007, LG = 19, R = 1
     << LG, MOD = 1000000007, MOD2 = 1e9 + 9, B = 256;
const ll INFLL = 1e18 + 7;

struct coord {
    int x, y, z;
    coord() {}
    coord(int a, int b, int c) : x(a), y(b), z(c) {}
    coord operator+(const coord o) const { return coord(x + o.x, y + o.y,
     z + o.z); }
    int operator*(const coord o) { return max({abs(x - o.x), abs(y - o.y)
    , abs(z - o.z)}); }
};
```

```
signed main() {
    fastIO;
    int n; cin >> n;
    coord b, e;
    cin >> b.x >> b.y >> b.z >> e.x >> e.y >> e.z;
    vector <coord> c(7);
    c[0] = coord(0, 0, 0);
    c[1] = coord(n, n, -2 * n);
    c[2] = coord(-n, -n, 2 * n);
    c[3] = coord(n, -2 * n, n);
    c[4] = coord(-n, 2 * n, -n);
    c[5] = coord(-2 * n, n, n);
    c[6] = coord(2 * n, -n, -n);
    int ans = INF;
    FOR(i, 0, 7) ans = min(ans, b * (c[i] + e));
    cout << ans;
}
```

Task C

```
#include <bits/stdc++.h>
// tiom4eg's precompiler options
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
// IO settings
#define fastIO ios_base::sync_with_stdio(false); cin.tie(0)
// Quick types
#define ll long long
#define ld long double
#define ull unsigned long long
#define pii pair <int, int>
#define vi vector <int>
#define mi vector <vector <int>>
// Quick functions
#define endl "\n"
#define F first
#define S second
#define all(a) a.begin(), a.end()
#define sz(a) (int)(a.size())
#define pb push_back
#define mp make_pair
// Quick fors
#define FOR(i, a, b) for (int i = a; i < b; ++i)
#define FORS(i, a, b, c) for (int i = a; i < b; i += c)
#define RFOR(i, a, b) for (int i = a; i >= b; --i)
#define EACH(e, a) for (auto& e : a)
// Pragmas
#ifndef TIOM4EG
#pragma GCC optimize("O3") // let the chaos begin!
#pragma GCC target("avx,avx2,tune=native")
#pragma GCC comment(linker, "/stack:200000000")
#endif
// PBDS
```

```cpp
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#define pbds tree <int, null_type, less <int>, rb_tree_tag,
    tree_order_statistics_node_update>
using namespace __gnu_pbds;
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
using namespace std;
mt19937 rng(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
//#define int long long
constexpr int INF = 1e9 + 7, MD = 998244353, MAX = 400007, LG = 19, R = 1
     << LG, MOD = 1000000007, MOD2 = 1e9 + 9, B = 256;
const ll INFLL = 1e18 + 7;


template <int MOD>
struct mint {
    int v;
    mint() : v(0) {}
    mint(int x) : v(x) {}
    mint operator+(const mint o) const { return mint(v + o.v - MOD * (v +
     o.v >= MOD)); }
    mint operator-(const mint o) const { return mint(v - o.v + MOD * (v <
     o.v)); }
    bool operator<(const mint o) const { return v < o.v; }
    bool operator==(const mint o) const { return v == o.v; }
    mint operator*(const mint o) const { return mint(v * 1LL * o.v % MOD)
    ; }
};

constexpr int MODS[3] = {1000000007, 1000000009, 1000000021};
using mint0 = mint <MODS[0]>;
using mint1 = mint <MODS[1]>;
using mint2 = mint <MODS[2]>;

mint0 b0 = mint0(3);
mint1 b1 = mint1(4);
mint2 b2 = mint2(5);

struct multihash {
    vector <mint0> p0;
    vector <mint1> p1;
    vector <mint2> p2;
    vector <mint0> h0;
    vector <mint1> h1;
    vector <mint2> h2;
    int n;
    multihash(string& s) {
        n = sz(s);
        p0.resize(n + 1), p1.resize(n + 1), p2.resize(n + 1);
        h0.resize(n + 1), h1.resize(n + 1), h2.resize(n + 1);
        p0[0] = mint0(1), p1[0] = mint1(1), p2[0] = mint2(1);
        FOR(i, 0, n) {
```

```
            p0[i + 1] = p0[i] * b0;
            p1[i + 1] = p1[i] * b1;
            p2[i + 1] = p2[i] * b2;
            h0[i + 1] = h0[i] * b0 + mint0(s[i] - '0' + 1);
            h1[i + 1] = h1[i] * b1 + mint1(s[i] - '0' + 1);
            h2[i + 1] = h2[i] * b2 + mint2(s[i] - '0' + 1);
        }
    }
    tuple <mint0, mint1, mint2> get_shift(int k) {
        mint0 r0 = h0[n] - h0[k] * p0[n - k];
        mint1 r1 = h1[n] - h1[k] * p1[n - k];
        mint2 r2 = h2[n] - h2[k] * p2[n - k];
        mint0 l0 = h0[k];
        mint1 l1 = h1[k];
        mint2 l2 = h2[k];
        return make_tuple(r0 * p0[k] + l0, r1 * p1[k] + l1, r2 * p2[k] +
    l2);
    }
};

#undef endl

signed main() {
    fastIO;
    string run; cin >> run;
    if (run[0] == 'f') {
        int n; cin >> n;
        string s; cin >> s;
        multihash mh(s);
        auto hsh = mh.get_shift(0);
        cout << get<0>(hsh).v << ' ' << get<1>(hsh).v << ' ' << get<2>(
    hsh).v << endl;
    }
    else {
        int n; cin >> n;
        string s; cin >> s;
        multihash mh(s);
        mint0 pr0; cin >> pr0.v;
        mint1 pr1; cin >> pr1.v;
        mint2 pr2; cin >> pr2.v;
        tuple <mint0, mint1, mint2> pr = make_tuple(pr0, pr1, pr2);
        FOR(i, 0, n) {
            if (mh.get_shift(i) == pr) return !(cout << "YES" << endl);
        }
        cout << "NO" << endl;
    }
}
```

Task D

```
#include <bits/stdc++.h>
// tiom4eg's precompiler options
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
```

```
// IO settings
#define fastIO ios_base::sync_with_stdio(false); cin.tie(0)
// Quick types
#define ll long long
#define ld long double
#define ull unsigned long long
#define pii pair <int, int>
#define vi vector <int>
#define mi vector <vector <int>>
// Quick functions
#define endl "\n"
#define F first
#define S second
#define all(a) a.begin(), a.end()
#define sz(a) (int)(a.size())
#define pb push_back
#define mp make_pair
// Quick fors
#define FOR(i, a, b) for (int i = a; i < b; ++i)
#define FORS(i, a, b, c) for (int i = a; i < b; i += c)
#define RFOR(i, a, b) for (int i = a; i >= b; --i)
#define EACH(e, a) for (auto& e : a)
// Pragmas
#ifndef TIOM4EG
#pragma GCC optimize("O3") // let the chaos begin!
//#pragma GCC target("avx,avx2,tune=native")
#pragma GCC comment(linker, "/stack:200000000")
#endif
// PBDS
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#define pbds tree <int, null_type, less <int>, rb_tree_tag,
    tree_order_statistics_node_update>
using namespace __gnu_pbds;
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
using namespace std;
mt19937 rng(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
//#define int long long
constexpr int INF = 1e9 + 7, MD = 998244353, MAX = 507, LG = 19, R = 1 <<
     LG, MOD = 1000000007, MOD2 = 1e9 + 9, B = 256;
const ll INFLL = 1e18 + 7;

#undef endl


// ? ? a ? ... x ? ? ... y
// move
// ? ? a x ... y ...
// shift
// ? a x ... y ...
// move
// ? a x y ...
```

```
// shift
// a x y ...
// move
// x a y
// shift
// a y
// here we know if x > y
// ...
// PROFIT!

int n;
char op;
deque <int> ans;

/*
void act(int p, int q) {
    deque <int> prv = ans;
    swap(ans[p], ans[q]);
    int x = ans.front();
    ans.pop_front();
    ans.push_back(x);
    //cout << "INTERACTOR STATE:";
    //FOR(i, 0, n) cout << ' ' << ans[i];
    //cout << endl;
    int fl = 1;
    FOR(i, 0, n) if (ans[i] != n - i) fl = 0;
    if (fl) {
        op = '!';
        cout << "MUST END HERE" << endl;
    }
    else {
        op = '=';
        FOR(i, 0, n) {
            if (prv[i] < ans[i]) {
                op = '>';
                break;
            }
            if (prv[i] > ans[i]) {
                op = '<';
                break;
            }
        }
    }
}
*/

int val[MAX];
deque <int> a;

void ask(int p, int q) {
    swap(a[p], a[q]);
    cout << p + 1 << ' ' << q + 1 << endl;
}
```

```
/*void ask(int p, int q) {
    swap(a[p], a[q]);
    act(p, q);
}*/
void shift() {
    int p = a.front();
    a.pop_front();
    a.pb(p);
    cin >> op;
    if (op == '!') exit(0);
}

int where(int x) { FOR(i, 0, n) if (a[i] == x) return i; }

signed main() {
    //fastIO;
    cin >> n;
    /*ans.resize(n);
    iota(all(ans), 1);
    shuffle(all(ans), rng);
    auto initial = ans;
    cout << "INITIAL STATE:";
    FOR(i, 0, n) cout << ' ' << ans[i];
    cout << endl;*/
    if (n <= 5) {
        while (true) {
            int i = rng() % n, j = rng() % n;
            cout << i + 1 << ' ' << j + 1 << endl;
            char st; cin >> st;
            if (st == '!') break;
        }
        return 0;
    }
    FOR(i, 0, n) a.pb(i);
    FOR(i, 0, n) FOR(j, i + 1, n) {
        int p = where(i);
        ask(p, 3);
        shift();
        int q = where(j);
        ask(q, 3);
        shift();
        //assert(ans[1] == initial[i] && ans[2] == initial[j]);
        ask(0, 1);
        shift();
        if (op == '<') val[i]++;
        else val[j]++;
    }
    vi o(n);
    iota(all(o), 0);
    sort(all(o), [&](int i, int j){ return val[i] > val[j]; });
    FOR(i, 0, n) {
        int p = where(o[i]);
        ask(p, n - 1);
```

```
        shift();
    }
    assert(op == '!');
}
```

Task E

```
#include <bits/stdc++.h>
// tiom4eg's precompiler options
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
// IO settings
#define fastIO ios_base::sync_with_stdio(false); cin.tie(0)
// Quick types
#define ll long long
#define ld long double
#define ull unsigned long long
#define pii pair <int, int>
#define vi vector <int>
#define mi vector <vector <int>>
// Quick functions
#define endl "\n"
#define F first
#define S second
#define all(a) a.begin(), a.end()
#define sz(a) (int)(a.size())
#define pb push_back
#define mp make_pair
// Quick fors
#define FOR(i, a, b) for (int i = a; i < b; ++i)
#define FORS(i, a, b, c) for (int i = a; i < b; i += c)
#define RFOR(i, a, b) for (int i = a; i >= b; --i)
#define EACH(e, a) for (auto& e : a)
// Pragmas
#ifndef TIOM4EG
#pragma GCC optimize("O3") // let the chaos begin!
//#pragma GCC target("avx,avx2,tune=native")
#pragma GCC comment(linker, "/stack:200000000")
#endif
// PBDS
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#define pbds tree <int, null_type, less <int>, rb_tree_tag,
    tree_order_statistics_node_update>
using namespace __gnu_pbds;
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
using namespace std;
mt19937 rng(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
#define int long long
constexpr int INF = 1e9 + 7, MD = 998244353, MAX = 100007, LG = 19, R = 1
     << LG, MOD = 1000000007, MOD2 = 1e9 + 9, B = 256;
const ll INFLL = 1e18 + 7;
```

```
vi g[MAX];
int sh[MAX], ans[MAX];
int pos[5000000];
vector <pii> qs;


template <int MOD>
struct mint {
    int v;
    mint() : v(0) {}
    mint(int x) : v(x) {}
    mint operator+(const mint o) const { return mint(v + o.v - MOD * (v +
     o.v >= MOD)); }
    mint operator-(const mint o) const { return mint(v - o.v + MOD * (v <
     o.v)); }
    bool operator<(const mint o) const { return v < o.v; }
    bool operator==(const mint o) const { return v == o.v; }
    mint operator*(const mint o) const { return mint(v * 1LL * o.v % MOD)
    ; }
};

using m107 = mint <1000000007>;
using m109 = mint <1000000009>;

m107 B1(100007);
m109 B2(100009);

m107 p1[MAX];
m109 p2[MAX];

m107 h1;
m109 h2;

inline ll combine(m107 a, m109 b) { return a.v * 2147483647LL + b.v; }

// it looks like our sequence is cyclic with relatively small period
// let's hash (shift_0, ..., shift_{n-1}, where)
unordered_map <ll, int> seen;

signed main() {
    p1[0] = m107(1), p2[0] = m109(1);
    FOR(i, 0, MAX) p1[i + 1] = p1[i] * B1, p2[i + 1] = p2[i] * B2;
    fastIO;
    int n, q; cin >> n >> q;
    FOR(i, 0, n) {
        int d; cin >> d;
        FOR(j, 0, d) {
            int v; cin >> v, --v;
            g[i].pb(v);
        }
    }
    FOR(i, 0, q) {
        int t; cin >> t;
```

```
        qs.pb({t, i});
    }
    sort(all(qs));
    int ptr = 0, now = 0;
    FOR(i, 0, n + 1) h1 = h1 + p1[i];
    FOR(i, 0, n + 1) h2 = h2 + p2[i];
    // old
    if (n > 1000) {
        for (int now = 0, where = 0; ptr < q; ++now) {
            if (qs[ptr].F == now) ans[qs[ptr].S] = where, ptr++;
            int nx = g[where][sh[where]];
            sh[where] = (sh[where] == sz(g[where]) - 1 ? 0 : sh[where] +
   1), where = nx;
        }
        FOR(i, 0, q) cout << ans[i] + 1 << ' ';
        return 0;
    }
    // new
    int preperiod, period;
    for (int now = 0, where = 0; ; ++now) {
        pos[now] = where;
        if (seen.find(combine(h1, h2)) != seen.end()) {
            preperiod = seen[combine(h1, h2)], period = now - seen[
   combine(h1, h2)];
            break;
        }
        seen[combine(h1, h2)] = now;

        h1 = h1 - m107(sh[where] + 1) * p1[where];
        h2 = h2 - m109(sh[where] + 1) * p2[where];
        h1 = h1 - m107(where + 1) * p1[n];
        h2 = h2 - m109(where + 1) * p2[n];
        int nx = g[where][sh[where]];
        sh[where] = (sh[where] == sz(g[where]) - 1 ? 0 : sh[where] + 1);
        h1 = h1 + m107(sh[where] + 1) * p1[where];
        h2 = h2 + m109(sh[where] + 1) * p2[where];
        where = nx;
        h1 = h1 + m107(where + 1) * p1[n];
        h2 = h2 + m109(where + 1) * p2[n];
    }
    FOR(i, 0, q) {
        if (qs[i].F < preperiod) ans[qs[i].S] = pos[qs[i].F];
        else ans[qs[i].S] = pos[preperiod + (qs[i].F - preperiod) %
   period];
    }
    FOR(i, 0, q) cout << ans[i] + 1 << ' ';
}
//
```

Task F

```
#include <bits/stdc++.h>
// tiom4eg's precompiler options
```

```
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
// IO settings
#define fastIO ios_base::sync_with_stdio(false); cin.tie(0)
// Quick types
#define ll long long
#define ld long double
#define ull unsigned long long
#define pii pair <int, int>
#define vi vector <int>
#define mi vector <vector <int>>
// Quick functions
#define endl "\n"
#define F first
#define S second
#define all(a) a.begin(), a.end()
#define sz(a) (int)(a.size())
#define pb push_back
#define mp make_pair
// Quick fors
#define FOR(i, a, b) for (int i = a; i < b; ++i)
#define FORS(i, a, b, c) for (int i = a; i < b; i += c)
#define RFOR(i, a, b) for (int i = a; i >= b; --i)
#define EACH(e, a) for (auto& e : a)
// Pragmas
#ifndef TIOM4EG
#pragma GCC optimize("O3") // let the chaos begin!
//#pragma GCC target("avx,avx2,tune=native")
#pragma GCC comment(linker, "/stack:200000000")
#endif
// PBDS
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#define pbds tree <int, null_type, less <int>, rb_tree_tag,
    tree_order_statistics_node_update>
using namespace __gnu_pbds;
// POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS POGGERS
using namespace std;
mt19937 rng(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
//#define int long long
constexpr int INF = 1e9 + 7, MD = 998244353, MAX = 57, LG = 19, R = 1 <<
    LG, MOD = 1000000007, MOD2 = 1e9 + 9, B = 256;
const ll INFLL = 1e18 + 7;

int h, w, a[MAX][MAX];
int cu(int r) { return (r + h - 1) % h; }
int cl(int c) { return (c + w - 1) % w; }

signed main() {
    fastIO;
    cin >> h >> w;
    vector <string> ans(h, string(w, '.'));
    FOR(i, 0, h) {
```

```
        string s; cin >> s;
        FOR(j, 0, w) a[i][j] = s[j] - '0';
    }
    int sum = 0;
    while (true) {
        int best = -1, bestx, besty, bestd;
        FOR(x, 0, h) FOR(y, 0, w) if (ans[x][y] == '.') {
            int nx = cu(x);
            if (ans[nx][y] == '.') {
                if (a[x][y] * a[nx][y] > best) best = a[x][y] * a[nx][y],
 bestx = x, besty = y, bestd = 0;
            }
            int ny = cl(y);
            if (ans[x][ny] == '.') {
                if (a[x][y] * a[x][ny] > best) best = a[x][y] * a[x][ny],
 bestx = x, besty = y, bestd = 1;
            }
        }
        if (best < 0) break;
        sum += best;
        ans[bestx][besty] = (bestd ? 'R' : 'B');
        if (bestd) ans[bestx][cl(besty)] = 'L';
        else ans[cu(bestx)][besty] = 'T';
    }
    cout << sum << endl;
    FOR(i, 0, h) cout << ans[i] << endl;
}
```