Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0267 – Лещинский Артём Владиславович

A	B	C	D	E	F	sum
100	100	100	100	100	82	582

Task A

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

int main() {
    cin.tie(0)->sync_with_stdio(0);
    int a, b, n;
    cin >> a >> b >> n;
    vector<ll> dp(n + 1);
    dp[1] = 1;
    for (int i = 2; i <= n; ++i) {
        if (i >= a)
            dp[i] += dp[i - a];
        if (i >= b) {
            dp[i] += dp[i - b];
        }
    }
    for (int i = 1; i <= n; ++i) {
        debug << dp[i] << ' ';
    }
    debug << endl;
```

```
    ll ans = 0;
    if (n - a + 1 >= 0)
        ans += dp[n - a + 1] * a;
    if (n - b + 1 >= 0)
        ans += dp[n - b + 1] * b;
    cout << ans;
    return 0;
}
```

Task B

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

template<typename T>
bool cmin(T& a, const T& b) {
    if (b < a) {
        a = b;
        return true;
    }
    return false;
}

vector<vector<int>> deltas;

inline int sign(int x) { return x < 0 ? -1 : 1; }

ll dist(ll x, ll y) {
    if (y > x) swap(x ,y);
    ll diag = 0;
    if (sign(x) != sign(y)) {
        assert(x >= 0 && y < 0);
        diag = min(x, -y);
        x -= diag;
        y += diag;
    }
    return diag + abs(x) + abs(y);
}

ll dist(int x1, int y1, int z1, int x2, int y2, int z2) {
    debug << x1 << ' ' << y1 << ' ' << z1 << " -> " << x2 << ' ' << y2 <<
     ' ' << z2 << endl;
    return dist(x2 - x1, y2 - y1);
}
```

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

    int x1, y1, z1, x2, y2, z2;
    cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;

    deltas = {
    //  {x, y, z}
        {-n, -n, 2 * n},
        {n, n, -2 * n},
    };

    ll ans = dist(x1, y1, z1, x2, y2, z2);
    vector<int> perm(3);
    iota(in(perm), 0);
    do {
        for (int t = 0; t < 2; ++t) {
            vector<int> coords = {x2, y2, z2};
            for (int i = 0; i < 3; ++i) {
                coords[perm[i]] += deltas[t][i];
            }
            cmin(ans, dist(x1, y1, z1, coords[0], coords[1], coords[2]));
        }
    } while (next_permutation(in(perm)));
    cout << ans;
    return 0;
}
```

Task C

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

template<typename T>
bool cmin(T& a, const T& b) {
    if (b < a) {
        a = b;
        return true;
    }
    return false;
}
```

```
typedef unsigned long long ull;

typedef tuple<ll, ull, ll> Hash;

const int MOD1 = (1LL << 31) - 1;
const ll MOD3 = 3;

Hash norm(Hash h) {
    get<0>(h) = (get<0>(h) >> 31) + (get<0>(h) & MOD1);
    get<0>(h) = (get<0>(h) >> 31) + (get<0>(h) & MOD1);
    get<2>(h) = (get<2>(h) % MOD3 + MOD3) % MOD3;
    return h;
}

Hash add(const Hash& h1, const Hash& h2) {
    return norm(Hash(get<0>(h1) + get<0>(h2),
        get<1>(h1) + get<1>(h2),
        get<2>(h1) + get<2>(h2)));
}

Hash sub(const Hash& h1, const Hash& h2) {
    return norm(Hash(get<0>(h1) - get<0>(h2),
        get<1>(h1) - get<1>(h2),
        get<2>(h1) - get<2>(h2)));
}

Hash mul(const Hash& h1, const Hash& h2) {
    return norm(Hash(get<0>(h1) * get<0>(h2),
        get<1>(h1) * get<1>(h2),
        get<2>(h1) * get<2>(h2)));
}

const Hash P(13, 1e9 + 7, 17);
const int mxN = 1e5 + 500;
Hash powp[mxN];

void precalc() {
    powp[0] = Hash(1, 1, 1);
    for (int i = 1; i < mxN; ++i) {
        powp[i] = mul(powp[i - 1], P);
    }
}

void solve1() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    if (n < 100) {
        cout << s << endl;
        return;
    }
```

```
    mt19937 rnd((n + 31) * 5 ^ 1948573948);
    Hash h(0, 0, 0);
    for (int i = 0; i < n; ++i) {
        int c = s[i] - '0' + 1;
        h = add(mul(h, P), Hash(c, c, c));
    }
    int cnt = count(in(s), '1');
    cout << get<0>(h) << ' ' << get<1>(h) << ' ' << get<2>(h) << '\n';
}

void solve2() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    if (n < 100) {
        string t;
        cin >> t;
        bool ans = false;
        for (int i = 0; i < n; ++i) {
            rotate(t.begin(), t.begin() + 1, t.end());
            if (t == s) {
                ans = true;
            }
        }
        cout << vector<string>{"NO", "YES"}[ans] << '\n';
        return;
    }
    Hash key;
    {
        ll x;
        ull y;
        ll z;
        cin >> x >> y >> z;
        key = Hash(x, y, z);
    }
    s = s + s;
    vector<Hash> pref(2 * n + 1);
    Hash h(0, 0, 0);
    mt19937 rnd((n + 31) * 5 ^ 1948573948);
    pref[0] = h;
    for (int i = 0; i < 2 * n; ++i) {
        int c = s[i] - '0' + 1;
        h = add(mul(h, P), Hash(c, c, c));
        pref[i + 1] = h;
    }
    bool ans = false;
    for (int i = 0; i < n; ++i) {
        Hash h(0, 0, 0);
        // for (int j = 0; j < n; ++j) {
        //     int c = s[(i + j) % n] - '0' + 1;
        //     h = add(mul(h, P), Hash(c, c));
        // }
```

```
        h = sub(pref[i + n], mul(pref[i], powp[n]));
        if (h == key) {
            ans = true;
        }
    }
    cout << vector<string>{"NO", "YES"}[ans] << '\n';
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    precalc();
    string run;
    cin >> run;
    if (run == "first") {
        solve1();
    }
    else {
        solve2();
    }
    return 0;
}
```

Task D

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

template<typename T>
bool cmin(T& a, const T& b) {
    if (b < a) {
        a = b;
        return true;
    }
    return false;
}

typedef unsigned long long ull;

typedef tuple<ll, ull, ll> Hash;

int cnt = 0;
int n;

int norm(int x) {
```

```
    return (x % n + n) % n;
}

void swp(int i, int j) {
    debug << i << ' ' << j << endl;
    cout << norm(i - cnt) + 1 << ' ' << norm(j - cnt) + 1 << endl;
    char resp;
    cin >> resp;
    if (resp == '!') {
        exit(0);
    }
    cnt += 1;
}

int caret, me;

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n;
    while (true) {
        bool work = false;
        cout << "1 1" << endl;
        char resp;
        cin >> resp;
        if (resp == '!') exit(0);

        for (int i = 0; i + 1 < n; ++i) {
            if (resp == '>') {
                cout << "1 " << n << endl;
                work = true;
                cin >> resp;
            }
            else {
                cout << "1 1" << endl;
                cin >> resp;
            }
            if (resp == '!') exit(0);

        }
        if (!work) break;

        // cout << "1 1" << endl;
        // cin >> resp;
        if (resp == '!') exit(0);

    }
    for (int i = 0; i < n + 1; ++i) {
        cout <<"1 1" << endl;
        char resp;
        cin >> resp;
        if (resp == '!') exit(0);
    }
    return 0;
```

```
}
```

Task E

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

template<typename T>
bool cmin(T& a, const T& b) {
    if (b < a) {
        a = b;
        return true;
    }
    return false;
}

const int mxN = 1e5 + 500;
int n, q;
vector<int> g[mxN];

vector<int> ord;
bool rotated[mxN];
vector<int> que;
int pos[mxN], nxt[mxN];
int par[mxN];

struct CleverStruct {

    struct block {
        int sm;
    };
    vector<block> blocks;
    vector<int> a;
    const int K = 333;
    void init(int num) {
        a.resize(num);
        blocks.resize((num + K - 1) / K);
    }
    void set1(int ind) {
        a[ind] = 1;
        int bl = ind / K;
        blocks[bl].sm += 1;
    }
    int sum() {
```

```
        int res = 0;
        /*for (int i = 0; i < (int)a.size(); ++i) {
            res += a[i];
        }*/
        for (auto& b : blocks) res += b.sm;
        return res;
    }
    int get_ith_one(int k) {
        int cnt = 0;
        int i = 0;
        while (cnt + blocks[i / K].sm <= k) {
            cnt += blocks[i / K].sm;
            i += K;
        }
        for (; i < (int)a.size(); ++i) {
            if (cnt == k && a[i]) {
                return i;
            }
            cnt += a[i];
        }
        assert(false);
    }
} sd;

void dfs(int v, int p = - 1) {
    par[v] = p;
    int ind = find(in(g[v]), p) - g[v].begin();
    pos[v] = ord.size();
    ord.push_back(v);
    for (int i = ind + 1; i < (int)g[v].size(); ++i) {
        int u = g[v][i];
        dfs(u, v);
        nxt[u] = ord.size();
        ord.push_back(v);
    }
    for (int i = 0; i < min(ind, (int)g[v].size()); ++i) {
        int u = g[v][i];
        dfs(u, v);
        nxt[u] = ord.size();
        ord.push_back(v);
    }
}

void euler(int v, int p = -1) {
    // ord.push_back(v);
    sd.set1(pos[v]);
    if (p != -1) {
        sd.set1(nxt[v]);
    }
    for (int u : g[v]) {
        if (u == p) break;
        euler(u, v);
        //ord.push_back(v);
```

```
    }

    if (!rotated[v]) {
        auto it = find(in(g[v]), p);
        int beg = it - g[v].begin();
        for (++beg; beg < (int)g[v].size(); ++beg) {
            que.push_back(g[v][beg]);
        }
        if (it != g[v].end()) {
            rotate(g[v].begin(), ++it, g[v].end());
        }
        rotated[v] = true;
    }
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> q;
    for (int i = 0; i < n; ++i) {
        int k;
        cin >> k;
        for (int j = 0; j < k; ++j) {
            int v;
            cin >> v, --v;
            g[i].push_back(v);
        }
    }

    set<pair<ll, int>> to_answer;
    vector<int> ans(q);
    for (int qry = 0; qry < q; ++qry) {
        ll t;
        cin >> t;

        to_answer.emplace(t, qry);
    }

    fill(rotated, rotated + n, false);

    dfs(0, -1);

    sd.init(ord.size());
    ll l = 0;
    int prev_size = 0;
    que = {0};
    while (!to_answer.empty()) {
        // ord.clear();
        auto tmp = que; que.clear();
        for (int v : tmp) {
            euler(v, par[v]);
        }
        ll r = l + sd.sum() -1;
        while (true) {
```

```
            auto it = to_answer.lower_bound(pair<ll, int>(l, -1));
            if (it != to_answer.end() && it->first < r) {
                ans[it->second] = ord[sd.get_ith_one(it->first - l)]; //
    ord[it->first - l];
                to_answer.erase(it);
            }
            else break;
        }
        debug << l << ' ' << r << ' ' << to_answer.size() << ": " << endl
    ;
        // for (int x : ord) debug << x << ' ';
        // debug << '\n';
        if (sd.sum() == prev_size) {
            // ord.pop_back();
            break;
        }
        prev_size = sd.sum();
        l = r;
    }
    for (auto elem : to_answer) {
        ans[elem.second] = ord[sd.get_ith_one((elem.first - l) % (sd.sum
    () - 1))];
    }
    for (int i = 0; i < q; ++i) {
        cout << ans[i] + 1 << ' ';
    }
    cout << '\n';
    return 0;
}
```

Task F

```
#include <bits/stdc++.h>
using namespace std;

#ifdef DEBUG
ofstream debug("debug.txt");
#else
#define debug if(false) cerr
#endif

typedef long long ll;
#define in(x) x.begin(), x.end()

template<typename T>
bool cmin(T& a, const T& b) {
    if (b < a) {
        a = b;
        return true;
    }
    return false;
}
```

```
typedef unsigned long long ull;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
int n, m;
vector<vector<int>> a;

int norm(int x, int mod) {
    x %= mod;
    if (x < 0) x += mod;
    return x;
}

bool is_free(const vector<vector<int>>& f, int i, int j) {
    if (f[i][j] != -1) return false;
    if (a[i][j] == 0) return false;
    if (f[norm(i - 1, n)][j] == 1 || f[i][norm(j - 1, m)] == 0) {
        return false;
    }
    return true;
}

ll cost(const vector<vector<int>>& f) {
    ll res = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            if (f[i][j] == 0) {
                res += a[i][j] * a[i][(j + 1) % m];
            }
            if (f[i][j] == 1) {
                res += a[i][j] * a[(i + 1) % n][j];
            }
        }
    }
    return res;
}

void print(const vector<vector<int>>& f) {
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            if (f[i][j] == 0) {
                cout << "L";
            }
            else if(f[i][j] == 1) {
                cout << "T";
            }
            else if (f[norm(i - 1, n)][j] == 1) {
                cout << "B";
            }
            else if (f[i][norm(j - 1, m)] == 0) {
                cout << "R";
            }
            else {
                cout << ".";
```

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

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    a.assign(n, vector<int>(m));
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j)
        {
            char c;
            cin >> c, c-= '0';
            a[i][j] = c;
        }
    }
    vector<vector<int>> field(n, vector<int>(m, -1));
    ll ans = 0;

    double temp = 1000;
    while (clock() / (double)CLOCKS_PER_SEC < 1.9) {
        //vector<vector<int>> new_field(n, vector<int>(m, -1));
        auto new_field = field;
        int cnt = 0;
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                if (rnd() % 100 < 3) {
                    new_field[i][j] = -1;
                }
                cnt += new_field[i][j] == -1;
            }
        }

        for (int i = 0; i < 10 * cnt; ++i) {
            int tp = rnd() % 2;
            int x = rnd() % n;
            int y = rnd() % m;
            if (tp == 0) {
                if (is_free(new_field, x, y) &&
                    is_free(new_field, x, (y + 1) % m)) {
                        new_field[x][y] = 0;
                    }
            }
            if (tp == 1) {
                if (is_free(new_field, x, y) &&
                    is_free(new_field, (x + 1) % n, y)) {
                        new_field[x][y] = 1;
                    }
            }
        }
        ll new_cost = cost(new_field);
```

```
        if (new_cost > ans || rnd() % 1000 < temp) {
            ans = new_cost;
            field = new_field;
        }
        else {
            // do nothing
        }
        temp *= 0.99;
    }
    cout << ans << '\n';
    print(field);
    return 0;
}
```