Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0264 – Леденёв Михаил Алексеевич

A	B	C	D	E	F	sum
100	100	100	100	20	89	509

Task A

```
//#pragma GCC optimize("Ofast,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")
//#include <ext/pb_ds/assoc_container.hpp>
//using namespace __gnu_pbds;
//using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;

using namespace std;
using ll = long long;
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

inline void work() {
    int a, b, n;
    cin >> a >> b >> n;
    if (b > n && a > n) {
        cout << 0 << endc;
```

```
    } else if (b > n) {
        ll global = 0;
        for (int i = 0; i < n; ++i) {
            vector<ll> dp(n, 0);
            dp[i] = 1;
            for (int j = i+1; j < n; ++j) {
                if (j - a >= i) dp[j] += dp[j-a];
            }
            int posA = (i - a + n) % n;
            global += dp[posA];
        }
        cout << global << endc;
    } else {
        ll global = 0;
        for (int i = 0; i < n; ++i) {
            vector<ll> dp(n, 0);
            dp[i] = 1;
            for (int j = i+1; j < n; ++j) {
                if (j - a >= i) dp[j] += dp[j-a];
                if (j - b >= i) dp[j] += dp[j-b];
            }
            int posA = (i - a + n) % n;
            int posB = (i - b + n) % n;
            global += dp[posA] + dp[posB];
        }
        cout << global << endc;
    }
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
    while (t --> 0) work();
}
```

Task B

```
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")

using namespace std;
using ll = long long;
```

```
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

ll distSimple(ll x1, ll y1, ll z1, ll x2, ll y2, ll z2) {
    x1 -= x2;
    y1 -= y2;
    z1 -= z2;
    ll plus = 0;
    if (x1 > 0) plus += x1;
    if (y1 > 0) plus += y1;
    if (z1 > 0) plus += z1;
    return plus;
}

inline void work() {
    ll n;
    cin >> n;
    ll x1, y1, z1, x2, y2, z2;
    cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
    ll de = n;
    vector<ll> variants = {
        distSimple(x1, y1, z1, x2, y2, z2),
        distSimple(x1, y1, z1, x2 + de, y2 + de, z2 - 2*de),
        distSimple(x1, y1, z1, x2 - de, y2 - de, z2 + 2*de),
        distSimple(x1, y1, z1, x2 + de, y2 - 2*de, z2 + de),
        distSimple(x1, y1, z1, x2 - de, y2 + 2*de, z2 - de),
        distSimple(x1, y1, z1, x2 - 2*de, y2 + de, z2 + de),
        distSimple(x1, y1, z1, x2 + 2*de, y2 - de, z2 - de)
    };
    cout << (*min_element(variants.begin(), variants.end())) << endc;
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
    while (t --> 0) work();
}
```

Task C

```
//#pragma GCC optimize("Ofast,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
```

```
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")

using namespace std;
using ll = long long;
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

int n;
string what, key;

const ll mod = 1e9+7;
const ll k = 227;
vector<ll> pows;

void precalc() {
    pows.resize(201000);
    pows[0] = 1;
    for (int i = 1; i < pows.size(); ++i) {
        pows[i] = (pows[i-1] * k) % mod;
    }
}

struct Hash {
    vector<ll> values;
    Hash(string w) {
        values.resize(w.size()+1, 0);
        for (int i = 0; i < w.size(); ++i) {
            ll now = w[i] - '0';
            values[i+1] = (values[i] * k + now) % mod;
        }
    }
    ll get(int start, int len) {
        ll minus = (values[start] * pows[len]) % mod;
        return (values[start+len] - minus + mod) % mod;
    }
};

void first() {
    Hash h = Hash(what);
    cout << h.get(0, n);
}

void second() {
    what += what;
    Hash h = Hash(what);
    ll cmp = stoll(key);
```

```
    for (int i = 0; i < n; ++i) {
        if (h.get(i, n) == cmp) {
            cout << "YES";
            return;
        }
    }
    cout << "NO";
}

inline void work() {
    precalc();
    string type;
    cin >> type >> n >> what;
    if (type == "first") {
        first();
    } else {
        cin >> key;
        second();
    }
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
    while (t --> 0) work();
}
```

Task D

```
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")

using namespace std;
using ll = long long;
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

struct interactor {
```

```
    int n;
    interactor() {
        cin >> n;
    }
    int swap(int i = 0, int j = 0) {
        if (i < n) i += n;
        if (j < n) j += n;
        if (i >= n) i -= n;
        if (j >= n) j -= n;
        cout << i+1 << ' ' << j+1 << endl;
        char result;
        cin >> result;
        if (result == '<') return -1;
        if (result == '>') return 1;
        if (result == '!') return 0;
        exit(0);
    }
} it;

inline void work() {
    while (1) {
        int result = it.swap();
        for (int i = 0; i < it.n-1; ++i) {
            if (result > 0) {
                result = it.swap(0, it.n-1);
            } else {
                result = it.swap(0, 0);
            }
        }
    }
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
    while (t --> 0) work();
}
```

Task E

```
//#pragma GCC optimize("Ofast,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")
```

```
using namespace std;
using ll = long long;
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

int n, q, t = 0, collision;
vector<vector<int>> childs;
vector<int> starts;

deque<int> path;

void simple() {
    int v = 0;
    for (int i = 0; i <= 100000; ++i) {
        path.push_back(v);
        int to = childs[v][starts[v]];
        UPD : {
            starts[v] = starts[v]+1;
            if (starts[v] == childs[v].size()) starts[v] = 0;
        }
        v = to;
    }
}

inline void work() {
    cin >> n >> q;
    childs.resize(n);
    starts.resize(n, 0);
    for (int i = 0; i < n; ++i) {
        int r;
        cin >> r;
        childs[i].resize(r);
        for (auto& e : childs[i]) cin >> e, --e;
    }
    simple();
    for (int i = 0; i < q; ++i) {
        ll time;
        cin >> time;
        cout << path[time]+1 << endc;
    }
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
```

```
    while (t --> 0) work();
}

/*

6 9
2 3 2
2 4 1
3 1 5 6
1 2
1 3
1 3
0 1 2 3 4 5 6 7 8

*/
```

Task F

```
//#pragma GCC optimize("Ofast,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
//#include <immintrin.h>
#include <bits/stdc++.h>
#include <random>
//#pragma GCC comment(linker, "/stack:200000000")
//#pragma GCC optimize("Ofast,unroll-loops,fast-math,inline,no-stack-
    protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,avx,avx2,abm
    ,mmx,popcnt")

using namespace std;
using ll = long long;
using ull = unsigned long long;
using uint = unsigned int;
using ld = long double;
using lll = __int128;
using uns = unsigned short;
const char endc = '\n';

int w, h;

/*
 24fin0264
 thimanted
 */

string get1() {
    string what = "";
    for (int j = 0; j < w; ++j) {
        if (j & 1) {
            what += 'R';
        } else {
            what += 'L';
        }
```

```
    }
    return what;
}

string get2() {
    string what = "";
    for (int j = 0; j < w; ++j) {
        if (j & 1) {
            what += 'L';
        } else {
            what += 'R';
        }
    }
    return what;
}

vector<string> nums;

pair<ll, vector<string>> solve1() {
    ll answer = 0;
    vector<string> ans;
    for (int i = 0; i < h; ++i) {
        ll res1 = 0, res2 = 0;
        for (int j = 0; j < w; j += 2) res1 += (nums[i][j] - '0') * (nums
    [i][j+1] - '0');
        for (int j = 1; j < w; j += 2) res2 += (nums[i][j] - '0') * (nums
    [i][(j+1) % w] - '0');
        if (res1 > res2) {
            ans.push_back(get1());
            answer += res1;
        } else {
            ans.push_back(get2());
            answer += res2;
        }
    }
    return {answer, ans};
}

ll get(int i, int a, int b) {
    return ll(nums[i][a] - '0') * ll(nums[i][b] - '0');
}

pair<ll, vector<string>> solve2() {
    ll answer = 0;
    vector<string> ans;
    for (int i = 0; i < h; ++i) {
        pair<ll, string> t1 = {-1, ""}, t2 = {-1, ""}, t;
        T1 : {
            vector<pair<ll, string>> dp(w+1);
            dp[0] = {0, ""};
            dp[1] = {0, "."};
            for (int j = 2; j <= w; ++j) {
                pair<ll, string> v1 = {dp[j-2].first + get(i, j-1, j-2),
```

```
    dp[j-2].second + "LR"};
                pair<ll, string> v2 = {dp[j-1].first, dp[j-1].second + ".
    "};
                dp[j] = max(v1, v2);
            }
            t1 = dp.back();
        }
        T2 : {
            nums[i] = nums[i].substr(1, w-1) + nums[i].substr(0, 1);
            vector<pair<ll, string>> dp(w+1);
            dp[0] = {0, ""};
            dp[1] = {0, "."};
            for (int j = 2; j <= w; ++j) {
                pair<ll, string> v1 = {dp[j-2].first + get(i, j-1, j-2),
    dp[j-2].second + "LR"};
                pair<ll, string> v2 = {dp[j-1].first, dp[j-1].second + ".
    "};
                dp[j] = max(v1, v2);
            }
            t2 = dp.back();
            t2.second = t2.second.substr(w-1, 1) + t2.second.substr(0, w
    -1);
        }
        t = max(t1, t2);
        ans.push_back(t.second);
        answer += t.first;
    }
    return {answer, ans};
}

ll get(int i1, int j1, int i2, int j2) {
    return ll(nums[i1][j1] - '0') * ll(nums[i2][j2] - '0');
}

ll getCost(vector<string> from) {
    ll answer = 0;
    for (int i = 0; i < from.size(); ++i) {
        for (int j = 0; j < w; ++j) {
            if (from[i][j] == 'L') {
                answer += get(i, j, i, (j+1) % w);
            }
            if (from[i][j] == 'T') {
                answer += get(i, j, i+1, j);
            }
        }
    }
    return answer;
}

pair<ll, vector<string>> solve3() {
    vector<string> matrix(h, get1());
    ll current = getCost(matrix);
    for (int i = 0; i < w*h / 4; ++i) {
```

```
        int x = rand() % w;
        int y = rand() % (h-1);
        int nxt = (x + 1) % w;
        int nyt = y+1;
        if (matrix[y][x] == 'L' && matrix[y][nxt] == 'R' && matrix[nyt][x
    ] == 'L' && matrix[nyt][nxt] == 'R') {
            ll resA = get(y, x, nyt, x) + get(y, nxt, nyt, nxt);
            ll resB = get(y, x, y, nxt) + get(nyt, x, nyt, nxt);
            if (resA > resB) {
                matrix[y][x] = 'T';
                matrix[y][nxt] = 'T';
                matrix[y+1][x] = 'B';
                matrix[y+1][nxt] = 'B';
                current -= resB;
                current += resA;
            }
        }
    }
    return {current, matrix};
}

pair<ll, vector<string>> solve4() {
    vector<string> matrix(h, get2());
    ll current = getCost(matrix);
    for (int i = 0; i < w*h / 4; ++i) {
        int x = rand() % w;
        int y = rand() % (h-1);
        int nxt = (x + 1) % w;
        int nyt = y+1;
        if (matrix[y][x] == 'L' && matrix[y][nxt] == 'R' && matrix[nyt][x
    ] == 'L' && matrix[nyt][nxt] == 'R') {
            ll resA = get(y, x, nyt, x) + get(y, nxt, nyt, nxt);
            ll resB = get(y, x, y, nxt) + get(nyt, x, nyt, nxt);
            if (resA > resB) {
                matrix[y][x] = 'T';
                matrix[y][nxt] = 'T';
                matrix[y+1][x] = 'B';
                matrix[y+1][nxt] = 'B';
                current -= resB;
                current += resA;
            }
        }
    }
    return {current, matrix};
}

pair<ll, vector<string>> solve5(vector<string> matrix) {
    ll current = getCost(matrix);
    for (int i = 0; i < w*h / 4; ++i) {
        int x = rand() % w;
        int y = rand() % (h-1);
        int nxt = (x + 1) % w;
        int nyt = y+1;
```

```
        if ((matrix[y][x] == 'L' && matrix[y][nxt] == 'R' && matrix[nyt][
    x] == 'L' && matrix[nyt][nxt] == 'R') || (matrix[y][x] == '.' &&
    matrix[y][nxt] == '.' && matrix[nyt][x] == '.' && matrix[nyt][nxt] ==
    '.')) {
            ll resA = get(y, x, nyt, x) + get(y, nxt, nyt, nxt);
            ll resB = get(y, x, y, nxt) + get(nyt, x, nyt, nxt);
            if (resA > resB) {
                matrix[y][x] = 'T';
                matrix[y][nxt] = 'T';
                matrix[y+1][x] = 'B';
                matrix[y+1][nxt] = 'B';
                current -= resB;
                current += resA;
            }
        }
    }
    return {current, matrix};
}

pair<ll, vector<string>> greedy() {
    vector<pair<ll, pair<int, int>>> whats;
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            whats.push_back({nums[i][j]-'0', {i, j}});
        }
    }
    sort(whats.rbegin(), whats.rend(), [&] (pair<ll, pair<int, int>> a,
    pair<ll, pair<int, int>> b) {
        return a.first < b.first;
    });
    string refa = "";
    for (int i = 0; i < w; ++i) refa.push_back('.');
    vector<string> local(h, refa);
    ll ans = 0;
    for (auto eee : whats) {
        int x = eee.second.second, y = eee.second.first;
        if (local[y][x] == '.') {
            vector<pair<pair<ll, char>, pair<int, int>>> vars;
            if (local[y][(x+1) % w] == '.') {
                vars.push_back({{get(y, x, y, (x+1) % w), 'R'}, {y, (x+1)
     % w}});
            }
            if (local[y][(x-1+w) % w] == '.') {
                vars.push_back({{get(y, x, y, (x-1+w) % w), 'L'}, {y, (x
    -1+w) % w}});
            }
            if (y > 0 && local[y-1][x] == '.') {
                vars.push_back({{get(y, x, y-1, x), 'T'}, {y-1, x}});
            }
            if (y+1 < h && local[y+1][x] == '.') {
                vars.push_back({{get(y, x, y+1, x), 'B'}, {y+1, x}});
            }
            sort(vars.rbegin(), vars.rend(), [&] (pair<pair<ll, char>,
```

```
    pair<int, int>> a, pair<pair<ll, char>, pair<int, int>> b) {
                return a.first < b.first;
            });
            if (vars.size() == 0) continue;
            if (vars.front().first.second == 'L') local[y][x] = 'R';
            if (vars.front().first.second == 'R') local[y][x] = 'L';
            if (vars.front().first.second == 'T') local[y][x] = 'B';
            if (vars.front().first.second == 'B') local[y][x] = 'T';
            ans += vars.front().first.first;
            local[vars.front().second.first][vars.front().second.second]
    = vars.front().first.second;
        }
    }
    return {ans, local};
}

inline void work() {
    cin >> h >> w;
    nums.resize(h);
    for (int i = 0; i < h; ++i) cin >> nums[i];
    pair<ll, vector<string>> wew = greedy();
    for (int i = 0; i < 4500; ++i) wew = max(wew, solve3());
    for (int i = 0; i < 4500; ++i) wew = max(wew, solve4());
    vector<string> copy = nums;
    vector<string> strs = solve2().second;
    nums = copy;
    for (int i = 0; i < 5001; ++i) wew = max(wew, solve5(strs));
    wew = max(wew, solve2());
    cout << wew.first << endc;
    for (auto& e : wew.second) cout << e << endc;
}

signed main() {
    ios_base::sync_with_stdio(false), cin.tie(nullptr);
    cout.precision(10), cout << fixed;
    srand(int(time(NULL)));
    int t = 1;
    //cin >> t;
    while (t --> 0) work();
}
```