Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0119 – Григорян Сергей Геворгович

A	B	C	D	E	F	sum
100	100	100	35	20	50	405

Task A

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define all(x) (x).begin(), (x).end()
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
/*
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
```

```
using namespace std;
using ll = int64_t;
using ull = uint64_t;
using bigInt = __uint128_t;

const int N = 81;
ll c[N][N];
// c[k][n];

void precalc() {
    c[0][0] = 1;
    for (int i = 1; i < N; ++i) {
        c[0][i] = 1;
        c[1][i] = i;
    }
    for (int k = 2; k < N; ++k) {
        for (int i = k; i < N; ++i) {
            c[k][i] = c[k - 1][i - 1] + c[k][i - 1];
        }
    }
}

ll solve(int n, int a, int b) {
    ll res = 0;
    for (int i = 0; i * a <= n; ++i) {
        int j = (n - i * a) / b;
        if (j * b == n - i * a) {
            res += c[i][i + j];
        }
    }
    return res;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    precalc();
    ll a, b, n;
    cin >> a >> b >> n;
    ll res = 0;
    res = solve(n - a, a, b) * a + solve(n - b, a, b) * b;
    cout << res;
}
```

Task B

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define all(x) (x).begin(), (x).end()
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
```

```
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
/*
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
using namespace std;
using ll = int64_t;
using ull = uint64_t;

int dist(int x1, int y1, int z1, int x2, int y2, int z2) {
    return max({abs(x1 - x2), abs(y1 - y2), abs(z1 - z2)});
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    int x1, y1, z1;
    int x2, y2, z2;
    cin >> x1 >> y1 >> z1;
    cin >> x2 >> y2 >> z2;
    int res = dist(x1, y1, z1, x2, y2, z2);
    for (int y = 0, x, z; y <= n; ++y) {
        z = n - y;
        // x = abs(n)
        res = min(res, dist(x1, y1, z1, -n, y, z) + dist(x2, y2, z2, n, -
    z, -y));
        res = min(res, dist(x1, y1, z1, n, -z, -y) + dist(x2, y2, z2, -n,
     y, z));
        // y = abs(n)
        x = y;
        z = n - y;
        res = min(res, dist(x1, y1, z1, x, -n, z) + dist(x2, y2, z2, -z,
    n, -x));
        res = min(res, dist(x1, y1, z1, -z, n, -x) + dist(x2, y2, z2, x,
```

```
    -n, z));
        // z = abs(n)
        x = n - y;
        res = min(res, dist(x1, y1, z1, x, y, -n) + dist(x2, y2, z2, -y,
    -x, n));
        res = min(res, dist(x1, y1, z1, -y, -x, n) + dist(x2, y2, z2, x,
    y, -n));
    }
    cout << res;
}
```

Task C

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
#include <bitset>
/*
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
using namespace std;
using ll = int64_t;
using ull = uint64_t;

const int N = 1e5;
const int mod = (int)1e9 + 7;
const int p = 3;
ll pw[N + 1];
ull pwu[N + 1];

ll sum(ll a, ll b) {
```

```
    if (a + b < mod) return a + b;
    return a + b - mod;
}

ll dif(ll a, ll b) {
    if (a - b >= 0) return a - b;
    return a - b + mod;
}

ll mul(ll a, ll b) {
    return 1LL * a * b % mod;
}

struct hsh {
    pair<ll, ull> h;

    hsh() {
        h.first = h.second = 0;
    }

    hsh(int value) {
        h.first = h.second = value;
    }

    hsh(const ll& val1, const ull& val2) {
        h = {val1, val2};
    }

    hsh operator+(const hsh& other) const {
        return hsh(sum(h.first, other.h.first), h.second + other.h.second
    );
    }

    hsh operator*(const hsh& other) const {
        return hsh(mul(h.first, other.h.first), h.second * other.h.second
    );
    }

    hsh operator-(const hsh& other) const {
        return hsh(dif(h.first, other.h.first), h.second - other.h.second
    );
    }

    hsh operator*(const int& val) const {
        return hsh(mul(h.first, val), h.second * val);
    }

    bool operator==(const hsh& other) const {
        return h == other.h;
    }
};

hsh q[2 * N + 1];
```

```
void out_int(int val) {
    cout << bitset<32>(val);
}

void out_ull(ull val) {
    cout << bitset<64>(val);
}

hsh subs(int l, int r, hsh* t) {
    hsh res;
    res.h.first = dif(t[r].h.first, mul(t[l].h.first, pw[r - l]));
    res.h.second = t[r].h.second - t[l].h.second * pwu[r - l];
    return res;
}

ll to_ll(const string& s) {
    ll res = 0;
    for (char c : s) {
        res *= 2;
        res += c - '0';
    }
    return res;
}

ull to_ull(const string& s) {
    ull res = 0;
    for (char c : s) {
        res *= 2;
        res += c - '0';
    }
    return res;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    pw[0] = 1;
    pwu[0] = 1;
    for (int i = 0; i < N; ++i) {
        pw[i + 1] = pw[i] * p % mod;
        pwu[i + 1] = pwu[i] * p;
    }
    string t;
    cin >> t;
    if (t == "first") {
        int n;
        cin >> n;
        string s;
        cin >> s;
        hsh h;
        for (int i = 0; i < n; ++i) {
            h = h * p + hsh(s[i] - '0' + 1);
```

```
        }
        out_int(h.h.first);
        out_ull(h.h.second);
    } else {
        int n;
        cin >> n;
        string s;
        cin >> s;
        string h;
        cin >> h;
        hsh ht;
        ht.h.first = to_ll(h.substr(0, 32));
        ht.h.second = to_ull(h.substr(32));
        s += s;
        q[0] = hsh();
        for (int i = 0; i < 2 * n; ++i) {
            q[i + 1] = q[i] * p + hsh(s[i] - '0' + 1);
        }
        for (int i = 0; i < n; ++i) {
            if (ht == subs(i, i + n, q)) {
                cout << "YES\n";
                return 0;
            }
        }
        cout << "NO\n";
    }
}
```

Task D

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define all(x) (x).begin(), (x).end()
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
/*
#include <ext/pb_ds/assoc_container.hpp>
```

```
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
using namespace std;
using ll = int64_t;
using ull = uint64_t;

int st = 0;
const int N = 500;
int n;
int cur_p[N];
vector<int> g[N];
vector<int> lft[N];
bool used[N];
vector<int> t;
int pos[N];
int p[N];

int get(int i) {
    return find(cur_p, cur_p + n, i) - cur_p;
}

int query(int i, int j) {
    cout << i << " " << j << endl;
    swap(cur_p[i - 1], cur_p[j - 1]);
    char res;
    cin >> res;
    if (res == '<') return -1;
    if (res == '=') return 0;
    if (res == '>') return 1;
    exit(0);
}

void dfs(int v) {
    used[v] = true;
    for (int u : g[v]) {
        if (!used[u]) dfs(u);
    }
    t.push_back(v);
    pos[v] = t.size() - 1;
}

int main() {
    ios_base::sync_with_stdio(false);
    int n;
    cin >> n;
    vector<int> p(n);
    for (int i = 0; i < n; ++i) {
        p[i] = i + 1;
    }
```

```
    do {
        vector<int> q = p;
        for (int i = 0; i < n; ++i) {
            int j = find(all(q), n - i) - q.begin();
            query(j + 1, n - i);
            rotate(q.begin(), q.begin() + 1, q.end());
        }
    } while (next_permutation(all(p)));
}
```

Task E

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define all(x) (x).begin(), (x).end()
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
/*
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
using namespace std;
using ll = int64_t;
using ull = uint64_t;

const int N = 1e5;
int n, q;
vector<int> g[N];
int times[N];
int ans[N];
int st[N];

void dfs(int v, ll cur = 0, ll ptr = 0) {
```

```
    if (cur == times[ptr]) {
        ans[ptr] = v + 1;
        ++ptr;
    }
    if (ptr == q)
        return;
    int mv = st[v]++;
    if (st[v] == g[v].size()) st[v] = 0;
    dfs(g[v][mv], cur + 1, ptr);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cin >> n >> q;
    for (int i = 0; i < n; ++i) {
        int d;
        cin >> d;
        g[i].resize(d);
        for (int j = 0; j < d; ++j) {
            cin >> g[i][j];
            --g[i][j];
        }
    }
    for (int i = 0; i < q; ++i) {
        cin >> times[i];
    }
    dfs(0);
    for (int i = 0; i < q; ++i) {
        cout << ans[i] << " ";
    }
}
```

Task F

```
/*
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC target("avx2,tune=native")
*/
#define all(x) (x).begin(), (x).end()
#define mp(a, b) make_pair(a, b)
#define forn(i, j, n) for (int i = j; i < (int)n; ++i)
#define ford(i, j, z) for (int i = j; i > (int)z; --i)
#define eb emplace_back
#define debug(x) cout << (#x) << " = " << x << "\n";
#include <cstring>
#include <numeric>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstdint>
#include <map>
#include <set>
```

```
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
/*
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
template<class T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
    tree_order_statistics_node_update>;
*/
using namespace std;
using ll = int64_t;
using ull = uint64_t;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int h, w;
    cin >> h >> w;
    string t[h];
    for (int i = 0; i < h; ++i) cin >> t[i];
    ll res1[2] = {0}, res2[2] = {0};
    for (int k = 0; k < 2; ++k)
    for (int i = 0; i < h; ++i) {
        for (int j = k; j < w + k; j += 2) {
            res1[k] += (t[i][j] - '0') * (t[i][(j + 1) % w] - '0');
        }
    }
    for (int k = 0; k < 2; ++k)
    for (int j = 0; j < w; ++j) {
        for (int i = k; i < h + k; i += 2) {
            res2[k] += (t[i][j] - '0') * (t[(i + 1) % h][j] - '0');
        }
    }
    if (*max_element(res1, res1 + 2) > *max_element(res2, res2 + 2)) {
        int k = max_element(res1, res1 + 2) - res1;
        cout << res1[k] << "\n";
        for (int i = 0; i < h; ++i) {
            for (int j = 0; j < w; ++j) {
                if (j % 2 == k % 2) {
                    cout << 'L';
                } else {
                    cout << 'R';
                }
            }
            cout << "\n";
        }
    } else {
        int k = max_element(res2, res2 + 2) - res2;
        cout << res2[k] << "\n";
        for (int i = 0; i < h; ++i) {
            for (int j = 0; j < w; ++j) {
```

```
                if (i % 2 == k % 2) {
                    cout << 'T';
                } else {
                    cout << 'B';
                }
            }
            cout << "\n";
        }
    }
}
```