Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0523 – Маглыш Кирилл Андреевич

A	B	C	D	E	F	sum
100	100	100	100	20	88	508

Task A

```
#include <iostream>

using namespace std;
using ll = long long;

const ll MAX_N = 81;
ll dp[MAX_N][MAX_N];

ll res = 0;

void resolve() {
    ll a, b, n;
    cin >> a >> b >> n;
    for (ll i = 0; i <= n; ++i) {
        for (ll j = 0; j <= n; ++j) {
            if (i == 0 || j == 0) {
                dp[i][j] = 1;
                continue;
            }

            dp[i][j] = dp[i - 1][j] + dp[i][j - 1];
        }
    }

    for (ll aCnt = 0; aCnt <= 80; ++aCnt) {
        for (ll bCnt = 0; bCnt <= 80; ++bCnt) {
            if (aCnt * a + bCnt * b != n) {
                continue;
            }
```

```
            if (aCnt) {
                for (ll i = 0; i < a; ++i) {
                    res += dp[aCnt - 1][bCnt];
                }
            }

            if (bCnt) {
                for (ll i = 0; i < b; ++i) {
                    res += dp[aCnt][bCnt - 1];
                }
            }
        }
    }

    cout << res << '\n';
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    resolve();
    return 0;
}
```

Task B

```
#include <iostream>

using namespace std;
using ll = long long;

inline ll dist(ll x0, ll y0, ll z0, ll x1, ll y1, ll z1) {
    return (abs(x0 - x1) + abs(y0 - y1) + abs(z0 - z1)) / 2;
}

void resolve() {
    ll n;
    cin >> n;
    ll x0, y0, z0;
    ll x1, y1, z1;
    cin >> x0 >> y0 >> z0;
    cin >> x1 >> y1 >> z1;
    ll res = dist(x0, y0, z0, x1, y1, z1);

    ll Y = n;
    ll X = 0;
    ll Z = -n;
    for (ll i = 0; i <= n; ++i) {
        res = min(res, dist(x0, y0, z0, X, Y, Z) + dist(x1, y1, z1, X - n
    , Y - n, -Z));
        res = min(res, dist(x1, y1, z1, X, Y, Z) + dist(x0, y0, z0, X - n
```

```
    , Y - n, -Z));
        --Y;
        ++X;
    }

    Y = n;
    X = -n;
    Z = 0;
    for (ll i = 0; i <= n; ++i) {
        res = min(res, dist(x0, y0, z0, X, Y, Z) + dist(x1, y1, z1, X+n,
    -Y, Z+n));
        res = min(res, dist(x1, y1, z1, X, Y, Z) + dist(x0, y0, z0, X+n,
    -Y, Z+n));
        ++X;
        --Z;
    }

    Y = 0;
    X = -n;
    Z = n;
    for (ll i = 0; i <= n; ++i) {
        res = min(res, dist(x0, y0, z0, X, Y, Z) + dist(x1, y1, z1, -X, Y
    -n, Z-n));
        res = min(res, dist(x1, y1, z1, X, Y, Z) + dist(x0, y0, z0, -X, Y
    -n, Z-n));
        ++Y;
        --Z;
    }

    cout << res << '\n';
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    resolve();
    return 0;
}
```

Task C

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
using ll = long long;
using ull = unsigned long long;

const ll MAX_N = 2e5;

ll factor = 179;
```

```
ull powers1[MAX_N];
ll powers2[MAX_N];
const ll MOD = 1e9 + 7;

vector<int> createClasses(const string& src) {
    ll n = ll(src.size());
    int alpha = 2;
    vector<bool> used(alpha);
    for (const auto& item : src) {
        used[item - '0'] = true;
    }

    int id = -1;
    vector<int> ref(alpha);
    for (int i = 0; i < alpha; ++i) {
        if (used[i]) {
            ref[i] = ++id;
        }
    }

    vector<int> res(n << 1);
    for (int i = 0; i < n; ++i) {
        res[i] = ref[src[i] - '0'];
        res[i + n] = -1;
    }

    return res;
}

ll buildSufMas(const string& src) {
    ll n = ll(src.length());
    vector<int> classes = createClasses(src);
    vector<pair<pair<int, int>, int>> mas(n);
    for (int len = 1; len < n; len <<= 1) {
        for (int i = 0; i < n; ++i) {
            mas[i] = {{classes[i], classes[i + len]}, i};
        }

        sort(mas.begin(), mas.end());
        int cl = 0;
        for (int i = 0; i < n; ++i) {
            if (i && mas[i].first != mas[i - 1].first) {
                ++cl;
            }

            classes[mas[i].second] = cl;
        }
    }

    vector<int> sufMas(n);
    for (int i = 0; i < n; ++i) {
        sufMas[classes[i]] = i;
    }
```

```
    for (int i = 0; i < n; ++i) {
        if (sufMas[i] < n / 2) {
            return sufMas[i];
        }
    }

    return -1;
}

ull hashed1(const string& src) {
    ull H = 0;
    ll n = ll(src.length());
    for (int i = 0; i < n; ++i) {
        H += powers1[i] * (src[i] - '0');
    }

    return H;
}

ll hashed2(const string& src) {
    ll H = 0;
    ll n = ll(src.length());
    for (int i = 0; i < n; ++i) {
        H += (powers2[i] * (src[i] - '0')) % MOD;
    }

    return H;
}

void solveFirst() {
    ll n;
    cin >> n;
    string src;
    cin >> src;
    ll start = buildSufMas(src + src);
    src = src.substr(start) + src.substr(0, start);

    cout << hashed1(src) << ' ' << hashed2(src) << '\n';
}

void solveSecond() {
    ll n;
    cin >> n;
    string src;
    cin >> src;
    string tip1;
    string tip2;
    cin >> tip1 >> tip2;

    ll start = buildSufMas(src + src);
    src = src.substr(start) + src.substr(0, start);
```

```
    ull sHash1 = hashed1(src);
    ll sHash2 = hashed2(src);
    ull tHash1 = stoull(tip1);
    ll tHash2 = stoll(tip2);

    if (sHash1 == tHash1 && sHash2 == tHash2) {
        cout << "YES\n";
    } else {
        cout << "NO\n";
    }
}

void resolve() {
    string type;
    cin >> type;
    if (type == "first") {
        solveFirst();
    } else {
        solveSecond();
    }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    powers1[0] = 1;
    powers2[0] = 1;
    for (int i = 1; i < MAX_N; ++i) {
        powers1[i] = powers1[i - 1] * factor;
        powers2[i] = (powers2[i - 1] * factor) % MOD;
    }

    resolve();
    return 0;
}
```

Task D

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
using ll = long long;

const ll MAX_N = 500;

bool isLess[MAX_N][MAX_N];

ll n;

//string TEST = "25341";
```

```
/*bool ask(ll i, ll j) {
    cout << i + 1 << ' ' << j + 1 << endl;

    string A = TEST;
    swap(TEST[i], TEST[j]);
    TEST = TEST.substr(1, 4) + TEST[0];
    cout << TEST << endl;

    if (TEST == "54321") {
        cout << "!" << endl;
        exit(0);
    }

    return TEST < A;

    string res;
    cin >> res;
    if (res == "!") {
        exit(0);
    }

    return res == "<";
}*/

bool ask(ll i, ll j) {
    cout << i + 1 << ' ' << j + 1 << endl;

    string res;
    cin >> res;
    if (res == "!") {
        exit(0);
    }

    return res == "<";
}

void resolve() {
    cin >> n;
    vector<ll> posOf(n);
    for (ll i = 0; i < n; ++i) {
        posOf[i] = i;
    }

    for (int A = 0; A < n; ++A) {
        for (int B = A + 1; B < n; ++B) {
            ll prv = 0;
            for (ll j = 0; j < n; ++j) {
                if (posOf[j] == 1) {
                    prv = j;
                    break;
                }
            }
```

```
        ask(posOf[A], posOf[prv]);
        swap(posOf[A], posOf[prv]);
        for (ll i = 0; i < n; ++i) {
            posOf[i] = (posOf[i] - 1 + n) % n;
        }

        for (ll j = 0; j < n; ++j) {
            if (posOf[j] == 1) {
                prv = j;
                break;
            }
        }

        isLess[A][B] = !ask(posOf[B], posOf[prv]);
        isLess[B][A] = !isLess[A][B];
        swap(posOf[B], posOf[prv]);
        for (ll i = 0; i < n; ++i) {
            posOf[i] = (posOf[i] - 1 + n) % n;
        }
    }
}

vector<ll> ord(n);
for (int i = 0; i < n; ++i) {
    ord[i] = i;
}

sort(ord.begin(), ord.end(), [&](const ll i, const ll j) {
    return isLess[i][j];
});

reverse(ord.begin(), ord.end());
for (int i = 0; i < n; ++i) {
    ll other = 0;
    for (ll j = 0; j < n; ++j) {
        if (posOf[j] == 0) {
            other = j;
            break;
        }
    }

    ask(posOf[ord[i]], posOf[other]);
    swap(posOf[ord[i]], posOf[other]);
    for (ll j = 0; j < n; ++j) {
        posOf[j] = (posOf[j] - 1 + n) % n;
    }
}

for (int i = 0; i < n; ++i) {
    ask(0, 0);
}
```

```
    exit(10);
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    resolve();
    return 0;
}
```

Task E

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
using ll = long long;


void resolve() {
    ll n, q;
    cin >> n >> q;
    vector<vector<ll>> graph(n);
    vector<ll> P(n);
    vector<ll> way;
    for (ll i = 0; i < n; ++i) {
        ll sz;
        cin >> sz;
        graph[i].resize(sz);
        for (auto& item : graph[i]) {
            cin >> item;
            --item;
        }
    }

    ll V = 0;
    for (ll i = 0; i < 100005; ++i) {
        way.push_back(V);
        ll id = P[V];
        P[V] = (P[V] + 1) % ll(graph[V].size());
        V = graph[V][id];
    }

    while (q--) {
        ll t;
        cin >> t;
        cout << way[t] + 1 << ' ';
    }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    resolve();
```

```
    return 0;
}
```

Task F

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
using ll = long long;

pair<ll, vector<string>> hor(ll n, ll m, const vector<vector<ll>>& src) {
    vector<string> res(n);
    ll pts = 0;
    for (ll i = 0; i < n; ++i) {
        for (ll j = 0; j < m; j += 2) {
            res[i] += "LR";
            pts += src[i][j] * src[i][j + 1];
        }
    }

    return {pts, res};
}

pair<ll, vector<string>> ver(ll n, ll m, const vector<vector<ll>>& src) {
    vector<string> res(n);
    ll pts = 0;
    for (ll i = 0; i < n; i += 2) {
        for (ll j = 0; j < m; ++j) {
            res[i] += "T";
            res[i + 1] += "B";
            pts += src[i][j] * src[i + 1][j];
        }
    }

    return {pts, res};
}

struct Dom {
    ll val;
    ll I;
    ll J;
    bool hor;

    bool operator<(const Dom& other) const {
        return val < other.val;
    }
};

pair<ll, vector<string>> greed(ll n, ll m, const vector<vector<ll>>& src)
     {
    vector<Dom> doms;
```

```
    for (ll i = 0; i < n; ++i) {
        for (ll j = 0; j < m - 1; ++j) {
            doms.push_back({src[i][j] * src[i][j + 1], i, j, true});
        }
    }

    for (ll i = 0; i < n - 1; ++i) {
        for (ll j = 0; j < m; ++j) {
            doms.push_back({src[i][j] * src[i + 1][j], i, j, false});
        }
    }

    vector<string> res(n);
    for (ll i = 0; i < n; ++i) {
        for (ll j = 0; j < m; ++j) {
            res[i] += '.';
        }
    }

    sort(doms.rbegin(), doms.rend());
    ll pts = 0;
    for (const auto& item : doms) {
        if (item.hor) {
            if (res[item.I][item.J] == '.' && res[item.I][item.J + 1] ==
    '.') {
                pts += item.val;
                res[item.I][item.J] = 'L';
                res[item.I][item.J + 1] = 'R';
            }
        } else {
            if (res[item.I][item.J] == '.' && res[item.I + 1][item.J] ==
    '.') {
                pts += item.val;
                res[item.I][item.J] = 'T';
                res[item.I + 1][item.J] = 'B';
            }
        }
    }

    return {pts, res};
}

void resolve() {
    ll n, m;
    cin >> n >> m;
    vector<vector<ll>> src(n, vector<ll>(m));
    for (ll i = 0; i < n; ++i) {
        string inp;
        cin >> inp;
        for (ll j = 0; j < m; ++j) {
            src[i][j] = inp[j] - '0';
        }
    }
```

```
    vector<pair<ll, vector<string>>> ans(3);
    ans[0] = hor(n, m, src);
    ans[1] = ver(n, m, src);
    ans[2] = greed(n, m, src);
    sort(ans.begin(), ans.end());
    cout << ans.back().first << '\n';
    for (const auto& item : ans.back().second) {
        cout << item << '\n';
    }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    resolve();
    return 0;
}
```