Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0210 – Кобзев Андрей Михайлович

A	B	C	D	E	F	sum
100	100	100	75	20	91	486

Task A

```
a, b, n = map(int, input().split())
fact = [1]
for i in range(1, 81):
    fact.append(fact[-1] * i)
ans = 0
def C(n, k):
    return fact[n] // (fact[k] * fact[n - k])

for k1 in range(n + 1):
    for k2 in range(n + 1):
        if (k1 * a + k2 * b == n):
            ans += a * C(k1 + k2 - 1, k2)
            ans += b * C(k1 + k2 - 1, k1)
print(ans)
```

Task B

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

typedef int64_t ll;
typedef uint64_t ull;

#define int ll

const int inf = 1234567891;
const int M = 1e9 + 7;
const int INF = 1e17 + 19;
const double PI = acos(-1);
```

```
mt19937 rnd(time(0));

struct Point {
    int x, y, z;
};

int dist(Point a, Point b) {
    return (abs(a.x - b.x) + abs(a.y - b.y) + abs(a.z - b.z)) / 2;
}

bool check(time_t f, time_t s) {
    return (double)(s - f) / CLOCKS_PER_SEC >= 1.95;
}

void solve() {
    int n;
    cin >> n;
    int x, y, z;
    cin >> x >> y >> z;
    Point a = {x, y, z};
    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    Point b = {x1, y1, z1};
    int ans = dist(a, b);
    int mx = max({abs(b.x - a.x), abs(b.y - a.y), abs(b.z - a.z)});
    auto st = clock();
    if (mx == abs(b.z - a.z)) {
    // z = n
        for (int x2 = -n; x2 <= n; ++x2) {
            int z2 = n;
            int y2 = -(x2 + z2);
            Point c = {x2, y2, z2};
            int cur = dist(a, c);
            c.x += c.z;
            c.y += c.z;
            c.z *= -1;
            cur += dist(c, b);
            ans = min(ans, cur);
        }

        // z = -n
        for (int x2 = -n; x2 <= n; ++x2) {
            int z2 = -n;
            int y2 = -(x2 + z2);
            Point c = {x2, y2, z2};
            int cur = dist(a, c);
            c.x += c.z;
            c.y += c.z;
            c.z *= -1;
            cur += dist(c, b);
            ans = min(ans, cur);
        }
```

```
} else if (mx == abs(b.y - a.y)) {
    // y = n
    for (int x2 = -n; x2 <= n; ++x2) {
        int y2 = n;
        int z2 = -(x2 + y2);
        Point c = {x2, y2, z2};
        int cur = dist(a, c);
        c.x += c.y;
        c.z += c.y;
        c.y *= -1;
        cur += dist(c, b);
        ans = min(ans, cur);
    }

    // y = -n
    for (int x2 = -n; x2 <= n; ++x2) {
        int y2 = -n;
        int z2 = -(x2 + y2);
        Point c = {x2, y2, z2};
        int cur = dist(a, c);
        c.x += c.y;
        c.z += c.y;
        c.y *= -1;
        cur += dist(c, b);
        ans = min(ans, cur);
    }
} else {
    // x = n
    for (int y2 = -n; y2 <= n; ++y2) {
        int x2 = n;
        int z2 = -(x2 + y2);
        Point c = {x2, y2, z2};
        int cur = dist(a, c);
        c.y += c.x;
        c.z += c.x;
        c.x *= -1;
        cur += dist(c, b);
        ans = min(ans, cur);
    }

    // x = -n
    for (int y2 = -n; y2 <= n; ++y2) {
        int x2 = -n;
        int z2 = -(x2 + y2);
        Point c = {x2, y2, z2};
        int cur = dist(a, c);
        c.y += c.x;
        c.z += c.x;
        c.x *= -1;
        cur += dist(c, b);
        ans = min(ans, cur);
    }
}
```

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

signed main() {
    ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr)
    ;

    #ifdef kiquxd
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif

    int t = 1;
    // cin >> t;
    while (t--) {
        solve();
    }
}
```

Task C

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

typedef int64_t ll;
typedef uint64_t ull;

#define int ull

const int inf = 1234567891;
const int M = 1e9 + 7;
const int INF = 1e17 + 19;
const double PI = acos(-1);

mt19937 rnd(time(0));

const int p = 73;

void solve() {
    string tp;
    cin >> tp;
    int n;
    cin >> n;
    if (tp == "first") {
        string s;
        cin >> s;
        int h = 0;
        int h1 = 0;
        for (int i = 0; i < s.size(); ++i) {
            h = h * p + s[i];
            h1 = (h1 * p + s[i]) % M;
        }
        cout << h << ' ' << h1 << '\n';
```

```
    } else {
        string s;
        cin >> s;
        int hash;
        cin >> hash;
        int hash1;
        cin >> hash1;
        vector<int> h(2*n + 1), h1(2*n + 1);
        for (int i = 0; i < 2*n; ++i) {
            h[i + 1] = h[i] * p + s[i % n];
            h1[i + 1] = (h1[i] * p + s[i % n]) % M;
        }
        vector<int> pw(n + 5), pw1(n + 5);
        pw[0] = pw1[0] = 1;
        for (int i = 1; i <= n + 4; ++i) {
            pw[i] = pw[i - 1] * p;
            pw1[i] = pw1[i - 1] * p % M;
        }
        auto get = [&](int l, int r) {
            return h[r] - h[l - 1] * pw[r - l + 1];
        };
        auto get1 = [&](int l, int r) {
            return (h1[r] - h1[l - 1] * pw1[r - l + 1] % M + M) % M;
        };
        for (int i = n; i <= 2*n; ++i) {
            if (get(i - n + 1, i) == hash && get1(i - n + 1, i) == hash1)
    {
                cout << "YES\n";
                return;
            }
        }
        cout << "NO\n";
    }
}

signed main() {
    ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr)
    ;

    #ifdef kiquxd
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif

    int t = 1;
    // cin >> t;
    while (t--) {
        solve();
    }
}
```

Task D

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

typedef int64_t ll;
typedef uint64_t ull;

// #define int ll

const int inf = 1234567891;
const int M = 1e9 + 7;
const int INF = 1e17 + 19;
const double PI = acos(-1);

mt19937 rnd(time(0));

void solve() {
    int n;
    cin >> n;
    vector<int> p(n);
    vector<vector<int>> cmp(n, vector<int>(n));
    iota(p.begin(), p.end(), 0);
    for (int i = 0; i < n; ++i) {
        for (int pos = 0; pos < n - 1; ++pos) {
            int f = pos, s = pos + 1;
            cout << f + 1 << ' ' << s + 1 << endl;
            swap(p[f], p[s]);
            for (int it = 0; it < n; ++it) {
                if (it > 0) cout << "1 1" << endl;
                char sg;
                cin >> sg;
                if (sg == '!') return;
                if (pos == 0 || sg == '=') continue;
                int nxt = it + 1;
                if (nxt == n) nxt = 0;
                cmp[p[it]][p[nxt]] = (sg == '<');
                cmp[p[nxt]][p[it]] = (sg == '>');
            }
        }
    }
    deque<int> ord;
    vector<int> val(n, 1);
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (cmp[i][j]) ++val[i];
        }
        ord.push_back(val[i]);
    }
    for (int x = n - 1; x > 1; --x) {
        int pred = 0;
        for (int i = 0; i < n; ++i) {
            if (ord[i] == x + 1) pred = i;
        }
        pred = (pred + 1) % n;
```

```
        for (int i = 0; i < n; ++i) {
            if (ord[i] == x) {
                cout << i + 1 << ' ' << pred + 1 << endl;
                swap(ord[i], ord[pred]);
                char c;
                cin >> c;
                if (c == '!') return;
                break;
            }
        }
        ord.push_back(ord.front());
        ord.pop_front();
    }
    for (int i = 0; i < n; ++i) {
        cout << "1 1" << endl;
        char c;
        cin >> c;
        if (c == '!') return;
    }
    return;
}

signed main() {
    // ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(
    nullptr);

    #ifdef kiquxd
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif

    int t = 1;
    // cin >> t;
    while (t--) {
        solve();
    }
}
```

Task E

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

typedef int64_t ll;
typedef uint64_t ull;

#define int ull

const int inf = 1234567891;
const int M = 1e9 + 7;
const int INF = 1e17 + 19;
const double PI = acos(-1);
```

```
mt19937 rnd(time(0));

void solve() {
    int n, q;
    cin >> n >> q;
    vector<vector<int>> g(n);
    vector<int> shift(n);
    for (int i = 0; i < n; ++i) {
        int sz;
        cin >> sz;
        for (int j = 0; j < sz; ++j) {
            int v;
            cin >> v;
            g[i].push_back(--v);
        }
    }
    deque<int> tm(q);
    for (int i = 0; i < q; ++i) {
        cin >> tm[i];
    }
    int v = 0;
    for (int t = 0; tm.size() && t <= tm.back(); ++t) {
        if (t == tm[0]) {
            cout << v + 1 << ' ';
            tm.pop_front();
        }
        int S = g[v].size();
        int u = g[v][shift[v]];
        shift[v] = (shift[v] + 1) % S;
        v = u;
    }
    cout << '\n';
}

signed main() {
    ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr)
    ;

    #ifdef kiquxd
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif

    int t = 1;
    // cin >> t;
    while (t--) {
        solve();
    }
}
```

Task F

```
#include <bits/stdc++.h>
```

```
using namespace std;

typedef int64_t ll;
typedef uint64_t ull;

#define int ll

const int inf = 1234567891;
const int M = 1e9 + 7;
const int INF = 1e17 + 19;
const double PI = acos(-1);

mt19937 rnd(time(0));

void solve() {
    int n, m;
    cin >> n >> m;
    vector<vector<int>> a(n, vector<int>(m));
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            char c;
            cin >> c;
            a[i][j] = (c - '0');
        }
    }
    int sum = 0;
    vector<vector<char>> used(n, vector<char>(m, '.'));
    while (true) {
        int ans = 0;
        pair<int, int> pos1;
        int way = -1;
        // L - 1, D - 2
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                int c = rnd() % 2;
                if (c == 1) {
                    if (used[i][j] == '.' && used[(i + 1) % n][j] == '.'
    &&
                        ans < a[i][j] * a[(i + 1) % n][j]) {
                        ans = a[i][j] * a[(i + 1) % n][j];
                        pos1 = {i, j};
                        way = 2;
                    }
                    if (used[i][(j + 1) % m] == '.' && used[i][j] == '.'
    &&
                        ans < a[i][j] * a[i][(j + 1) % m]) {
                        ans = a[i][j] * a[i][(j + 1) % m];
                        pos1 = {i, j};
                        way = 1;
                    }
                } else {
                    if (used[i][(j + 1) % m] == '.' && used[i][j] == '.'
    &&
```

```
                        ans < a[i][j] * a[i][(j + 1) % m]) {
                        ans = a[i][j] * a[i][(j + 1) % m];
                        pos1 = {i, j};
                        way = 1;
                    }
                    if (used[i][j] == '.' && used[(i + 1) % n][j] == '.'
    &&
                        ans < a[i][j] * a[(i + 1) % n][j]) {
                        ans = a[i][j] * a[(i + 1) % n][j];
                        pos1 = {i, j};
                        way = 2;
                    }
                }
            }
        }
        if (way == -1) break;
        if (way == 1) {
            used[pos1.first][pos1.second] = 'L';
            used[pos1.first][(pos1.second + 1) % m] = 'R';
        } else {
            used[pos1.first][pos1.second] = 'T';
            used[(pos1.first + 1) % n][pos1.second] = 'B';
        }
        sum += ans;
    }
    cout << sum << '\n';
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            cout << used[i][j];
        }
        cout << '\n';
    }
}

signed main() {
    ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr)
    ;

    #ifdef kiquxd
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif

    int t = 1;
    // cin >> t;
    while (t--) {
        solve();
    }
}
```