Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0421 – Симонова Елизавета Игоревна

A	B	C	D	E	F	sum
100	25	40	100	20	91	376

Task A

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <deque>
#include <bitset>
#include <iomanip>
#include <queue>
#include <cmath>
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;

std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());
```

```
ll solve_otr(int a, int b, int n) {
    if (n < 0) {
        return 0;
    }
    vector<ll> dp(a + b + n + 1, 0);
    dp[0] = 1;
    for (int i = 0; i < n; ++i) {
        dp[i + a] += dp[i];
        dp[i + b] += dp[i];
    }
    return dp[n];
}

inline void solve() {
    int a, b, n;
    cin >> a >> b >> n;
    ll ans = solve_otr(a, b, n);
    for (int i = 1; i < n; ++i) {
        if (i < a) {
            ans += solve_otr(a, b, n - a);
        }
        if (i < b && a != b) {
            ans += solve_otr(a, b, n - b);
        }
    }
    cout << ans << '\n';
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```

Task B

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <deque>
#include <bitset>
#include <iomanip>
#include <queue>
#include <cmath>
```

```
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;

std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());

map<vector<int>, int> mp;
vector<vector<int>> ch = { {1, -1, 0}, {-1, 1, 0}, {1, 0, -1}, {-1, 0,
    1}, {0, 1, -1}, {0, -1, 1} };
vector<vector<int>> bs;

void bfs(int x, int y, int z, int n) {
    int dst = mp[{x, y, z}];
    for (auto p : ch) {
        int xn = x + p[0];
        int yn = y + p[1];
        int zn = z + p[2];
        if (abs(xn) > n || abs(yn) > n || abs(zn) > n) {
            xn = -x + p[0];
            yn = -y + p[1];
            zn = -z + p[2];
        }
        if (mp[{xn, yn, zn}] == 0) {
            if (abs(xn) == n || abs(yn) == n || abs(zn) == n) {
                mp[{-xn, -yn, -zn}] = dst + 1;
            }
            mp[{xn, yn, zn}] = dst + 1;
            bs.push_back({ xn, yn, zn });
        }
    }
}

inline void solve() {
    int n; cin >> n;
    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    int x2, y2, z2;
    cin >> x2 >> y2 >> z2;
    bs.push_back({ x1, y1, z1 });
    mp[{x1, y1, z1}] = 1;
```

```
    if (abs(x1) == n || abs(y1) == n || abs(z1) == n) {
        mp[{-x1, -y1, -z1}] = 1;
    }
    for (int i = 0; i < (int)bs.size(); ++i) {
        bfs(bs[i][0], bs[i][1], bs[i][2], n);
    }
    cout << mp[{x2, y2, z2}] - 1;
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```

Task C

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <deque>
#include <bitset>
#include <iomanip>
#include <queue>
#include <cmath>
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;

std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());

const ll P = 13;
```

```
inline void solve() {
    string s; cin >> s;
    if (s[0] == 'f') {
        int n; cin >> n;
        string s; cin >> s;
        cout << s;
        return;
    } else {
        int n; cin >> n;
        string s; cin >> s;
        string t; cin >> t;
        if (s.size() != t.size()) {
            while (true);
        }
        for (int i = 0; i < n; ++i) {
            bool y = true;
            for (int j = i; j < n + i; ++j) {
                if (s[j % n] != t[(j - i)]) {
                    y = false;
                }
            }
            if (y) {
                cout << "YES\n";
                return;
            }
        }
        cout << "NO\n";
        return;
    }
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```

Task D

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <deque>
#include <bitset>
```

```
#include <iomanip>
#include <queue>
#include <cmath>
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;

std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());

int sr[500][500];

bool comp(int a, int b) {
    if (sr[a][b] == 1) {
        return true;
    }
    return false;
}

inline void solve() {
    int n; cin >> n;
    vector<int> p(n);
    for (int i = 0; i < n; ++i) {
        p[i] = i;
    }
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            sr[i][j] = -1;
        }
    }
    int nw = 0;
    int nd = n * (n - 1) / 2;
    while (nw < nd) {
        int j = 1;
        while (j < n && sr[p[0]][p[j]] != -1) {
            ++j;
        }
        if (j == n) {
            cout << "1 1\n";
            cout.flush();
            char ch; cin >> ch;
```

```
            if (ch == '!') {
                return;
            }
        } else {
            cout << '2' << ' ' << j + 1 << '\n';
            cout.flush();
            char ch; cin >> ch;
            swap(p[1], p[j]);
            if (ch == '!') {
                return;
            }
            if (ch == '<') {
                sr[p[0]][p[1]] = 0;
                sr[p[1]][p[0]] = 1;
            } else {
                sr[p[0]][p[1]] = 1;
                sr[p[1]][p[0]] = 0;
            }
            ++nw;
        }
        vector<int> q(n);
        for (int i = 1; i < n; ++i) {
            q[i - 1] = p[i];
        }
        q[n - 1] = p[0];
        p = q;
    }
    vector<int> pr(n);
    for (int i = 0; i < n; ++i) {
        pr[i] = i;
    }
    sort(all(pr), comp);
    reverse(all(pr));
    for (int i = 1; i < n; ++i) {
        int f = 0;
        while (p[f] != pr[0]) {
            ++f;
        }
        int s = 0;
        while (p[s] != pr[i]) {
            ++s;
        }
        cout << (i + f) % n + 1 << ' ' << s + 1 << '\n';
        cout.flush();
        char ch; cin >> ch;
        if (ch == '!') {
            return;
        }
        swap(p[(i + f) % n], p[s]);
        vector<int> q(n);
        for (int i = 1; i < n; ++i) {
            q[i - 1] = p[i];
        }
```

```
        q[n - 1] = p[0];
        p = q;
    }
    while (true) {
        cout << "1 1\n";
        cout.flush();
        char ch; cin >> ch;
        if (ch == '!') {
            return;
        }
    }
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```

Task E

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <deque>
#include <bitset>
#include <iomanip>
#include <queue>
#include <cmath>
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;
```

```
std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());

vector<vector<int>> edge;
vector<int> NW;

inline void solve() {
    int n, q; cin >> n >> q;
    edge.resize(n);
    NW.resize(n, 0);
    for (int i = 0; i < n; ++i) {
        int d; cin >> d;
        for (int j = 0; j < d; ++j) {
            int v; cin >> v; --v;
            edge[i].push_back(v);
        }
    }
    int u = 0;
    vector<int> ans;
    for (int i = 0; i <= 100000; ++i) {
        ans.push_back(u + 1);
        int uw = u;
        u = edge[u][NW[u]];
        NW[uw] = (NW[uw] + 1) % (int)edge[uw].size();
    }
    vector<int> zp(q);
    for (int i = 0; i < q; ++i) {
        cin >> zp[i];
    }
    for (int i = 0; i < q; ++i) {
        cout << ans[zp[i]] << ' ';
    }
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```

Task F

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
```

```
#include <deque>
#include <bitset>
#include <iomanip>
#include <queue>
#include <cmath>
#include <ctime>
#include <numeric>
#include <string>
#include <chrono>
#include <random>

using namespace std;

typedef long long ll;
typedef long double ld;
#define all(x) x.begin(), x.end()
const ll MOD = 1000000007;
const ll inf = 1e18;
const ld eps = 1e-10;
ld pi = asin(0.5) * (ld)6;

std::mt19937 rnd(chrono::duration_cast<chrono::milliseconds>(chrono::
    system_clock::now().time_since_epoch()).count());

int tabl[50][50];
vector<vector<int>> v[100];
int ns[50][50];

inline void solve() {
    int h, w; cin >> h >> w;
    for (int i = 0; i < h; ++i) {
        string s; cin >> s;
        for (int j = 0; j < w; ++j) {
            tabl[i][j] = s[j] - '0';
        }
    }
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            int yn = (i + 1) % h;
            int xn = j;
            v[tabl[i][j] * tabl[yn][xn]].push_back({ i, j, yn, xn });
            yn = i;
            xn = (j + 1) % w;
            v[tabl[i][j] * tabl[yn][xn]].push_back({ i, j, yn, xn });
        }
    }
    int ans = 0;
    for (int i = 99; i >= 0; --i) {
        for (auto p : v[i]) {
            int y = p[0];
            int x = p[1];
            int xn = p[3];
            int yn = p[2];
```

```
            if (ns[y][x] == 0 && ns[yn][xn] == 0) {
                ans += i;
                if (y == yn) {
                    ns[y][x] = 1;
                    ns[yn][xn] = 2;
                } else {
                    ns[y][x] = 3;
                    ns[yn][xn] = 4;
                }
            }
        }
    }
    cout << ans << '\n';
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            if (ns[i][j] == 0) {
                cout << '.';
            } else if (ns[i][j] == 1) {
                cout << 'L';
            } else if (ns[i][j] == 2) {
                cout << 'R';
            } else if (ns[i][j] == 3) {
                cout << 'T';
            } else {
                cout << 'B';
            }
        }
        cout << '\n';
    }
}

signed main(void) {
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int t = 1; srand(time(0));
    while (t--) {
        solve();
    }
    return 0;
}
```