Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0062 – Брит Дмитрий Алексеевич

A	B	C	D	E	F	sum
100	100	40	100	20	3	363

Task A

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;

using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 100;
int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
   );
    int a, b, n; cin >> a >> b >> n;

    ll ans = 0;
    if (n >= a) {
```

```
        for (int st = 0; st < a; ++st) {
            vector<ll>dp(n + 1);
            dp[st] = 1;
            for (int i = st + 1; i <= n; ++i) {
                if (i - a >= 0) {
                    dp[i] += dp[i - a];
                }
                if (i - b >= 0) {
                    dp[i] += dp[i - b];
                }
            }
            //for (auto c : dp)cout << c << " "; cout << "\n";
            if (a - st >= 0 and n - (a - st) >= 0) {
                //cout << dp[n - (a - st)] << " a\n";
                ans += dp[n - (a - st)];
            }

        }
    }
    if (n >= b) {
        for (int st = 0; st < b; ++st) {
            vector<ll>dp(n + 1);
            dp[st] = 1;
            for (int i = st + 1; i <= n; ++i) {
                if (i - a >= 0) {
                    dp[i] += dp[i - a];
                }
                if (i - b >= 0) {
                    dp[i] += dp[i - b];
                }
            }
            //for (auto c : dp)cout << c << " "; cout << "\n";
            if (b - st >= 0 and n - (b - st) >= 0) {
                //cout << dp[n - (b - st)] << " b\n";
                ans += dp[n - (b - st)];
            }

        }
    }
    cout << ans;
}
//        : mandei
/*
1 2 1 1
1 1 1 2
1 1 2 1*/
```

Task B

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
```

```
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;

using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 100;
vector<vector<int>>g;
int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
    );
    /*int n; cin >> n;
    vector<vector<int>>g(n, vector<int>(n));
    for (int i = 0; i < n; ++i) {
        cout << "1 1" << endl;
        char x; cin >> x;
        if (x == '>') {
            g[i][i + 1] = 1;
        }
        else {
            g[i + 1][i] = 1;
        }
    }*/
    ll n, x, y, z, xx, yy, zz; cin >> n >> x >> y >> z >> xx >> yy >> zz;
    ll ans = (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2;
    zz += 2 * n, yy -= n, xx -= n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    zz -= 2 * n, yy += n, xx += n;
    zz -= 2 * n, yy += n, xx += n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    zz += 2 * n, yy -= n, xx -= n;

    xx += 2 * n, yy -= n, zz -= n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    xx -= 2 * n, yy += n, zz += n;
    xx -= 2 * n, yy += n, zz += n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    xx += 2 * n, yy -= n, zz -= n;

    yy += 2 * n, zz -= n, xx -= n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    yy -= 2 * n, zz += n, xx += n;
```

```
    yy -= 2 * n, zz += n, xx += n;
    ans = min(ans, (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2);
    yy += 2 * n, zz -= n, xx -= n;
    cout << ans;
}
//        : mandei
/*
1 2 1 1
1 1 1 2
1 1 2 1*/
```

Task C

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;

using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 100;
vector<vector<int>>g;
int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
   );
    /*int n; cin >> n;
    vector<vector<int>>g(n, vector<int>(n));
    for (int i = 0; i < n; ++i) {
        cout << "1 1" << endl;
        char x; cin >> x;
        if (x == '>') {
            g[i][i + 1] = 1;
        }
        else {
            g[i + 1][i] = 1;
        }
    }*/
    /*ll n, x, y, z, xx, yy, zz; cin >> n >> x >> y >> z >> xx >> yy >>
   zz;
```

```
    ll ans = (abs(x - xx) + abs(y - yy) + abs(z - zz)) / 2;
    int type1=0;
    if ((x == n and y == 0 and z == -n) or
        (x == 0 and y == -n and z == n) or
        (x == -n and y == n and z == 0)) {
        type1 = 1;
    }
    if ((x == n and y == -n and z == 0) or
        (x == 0 and y == n and z == -n) or
        (x == -n and y == 0 and z == n)) {
        type1 = -1;
    }
    int type2 = 0;
    if ((xx == n and yy == 0 and zz == -n) or
        (xx == 0 and yy == -n and zz == n) or
        (xx == -n and yy == n and zz == 0)) {
        type2 = 1;
    }
    if ((xx == n and yy == -n and zz == 0) or
        (xx == 0 and yy == n and zz == -n) or
        (xx == -n and yy == 0 and zz == n)) {
        type2 = -1;
    }

    bool cnttwo=
    if(abs(x)==n)
    cout << min(ans, (2ll * n + 1) - ans);*/



    string type; cin >> type;
    if (type == "first") {
        int n; string s; cin >> n >> s;
        cout << s << endl;
    }
    else {
        int n; string s, t; cin >> n >> s >> t;
        for (int i = 0; i < n; ++i) {
            string nw = t.substr(i,n-i)+t.substr(0, i);
            //cout << nw << "\n";
            if (nw == s) {
                cout << "YES";
                return 0;
            }
        }
        cout << "NO";
    }
}
//        : mandei
/*
1 2 1 1
1 1 1 2
1 1 2 1*/
```

Task D

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;

using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 1000000;
vector<int>a, cur_ind;
ll bl = 0;
void pass(){
    int x = a[0];
    for (int i = 0; i + 1 < a.size(); ++i) {
        a[i] = a[i + 1];
        cur_ind[a[i + 1]]--;
    }
    a[a.size() - 1] = x;
    cur_ind[x] = a.size() - 1;
}
char swap_ind(int v, int u) {
    cout << v + 1 << " " << u + 1 << endl;
    swap(a[u], a[v]);
    cur_ind[a[u]] = u;
    cur_ind[a[v]] = v;
    char c; cin >> c;
    pass();
    //for (auto c : a)cout << c << " "; cout << "a\n";
    //for (auto c : cur_ind)cout << c << " "; cout << "\n";
    if (c == '!')exit(0);
    return c;
}

int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
    );
    int n; cin >> n;
    a.resize(n), cur_ind.resize(n);
```

```
    for (int i = 0; i < n; ++i) {
        cur_ind[i] = i;
        a[i] = i;
    }
    vector<pair<int, int>>kw(n, { 0,0 });
    for (int i = 0; i < n; ++i)kw[i].second = i;
    for (int fs = 0; fs < n; ++fs) {
        //cout << fs<<"//\n";
        for (int ss = fs + 1; ss < n; ++ss) {
            if (cur_ind[fs] + cur_ind[ss] == 1) {
                char c = swap_ind(0, 0);
                if ((c == '<' and cur_ind[fs] == n-1) or (c == '>' and
    cur_ind[fs] == 0)) {
                    //cout << fs << " more than " << ss << "\n";
                    kw[fs].first++;
                }
                else {
                    //cout << ss << " more than " << fs << "\n";
                    kw[ss].first++;
                }
            }
            else if (cur_ind[fs] && cur_ind[ss] == 0) {
                char c = swap_ind(1, max(cur_ind[fs], cur_ind[ss]));
                if ((c == '<' and cur_ind[fs] == n-1) or (c == '>' and
    cur_ind[fs] == 0)) {
                    //cout << fs << " more than " << ss << "\n";
                    kw[fs].first++;
                }
                else {
                    //cout << ss << " more than " << fs << "\n";
                    kw[ss].first++;
                }
            }
            else {
                swap_ind(1, cur_ind[fs]);
                char c = swap_ind(1, cur_ind[ss]);
                if ((c == '<' and cur_ind[fs] == n-1) or (c == '>' and
    cur_ind[fs] == 0)) {
                    //cout << fs << " more than " << ss << "\n";
                    kw[fs].first++;
                }
                else {
                    //cout << ss << " more than " << fs << "\n";
                    kw[ss].first++;
                }
            }
        }
    }
    sort(kw.rbegin(), kw.rend());
    for (int i = 0; i < n; ++i) {
        swap_ind(cur_ind[kw[i].second], n - 1);
    }
}
```

```
//        : mandei
```

Task E

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;

using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 1000000;
vector<vector<int>>g;
int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
   );
    /*int n; cin >> n;
    vector<vector<int>>g(n, vector<int>(n));
    for (int i = 0; i < n; ++i) {
        cout << "1 1" << endl;
        char x; cin >> x;
        if (x == '>') {
            g[i][i + 1] = 1;
        }
        else {
            g[i + 1][i] = 1;
        }
    }*/

    int n, q; cin >> n >> q;
    ll period = (n - 1) * 2;
    g.resize(n);
    for (int v = 0; v < n; ++v) {
        int k; cin >> k;
        while (k--) {
            int u; cin >> u; --u;
            g[v].push_back(u);
        }
    }
```

```
    int v = 0, time = 0;
    vector<int>turn(n);
    vector<int>ans(maxn);
    while (time < ans.size()) {
        ans[time] = v;
        //cout << v+1 << " ";
        int aim = turn[v];
        turn[v] = (turn[v] + 1) % g[v].size();
        v = g[v][aim];
        time++;
    }
    //cout << "\n";
    int l = -1, r = maxn / 3;
    while (r - l > 1) {
        ll md = (r + l) / 2;
        bool gt = true;
        for (int i = 0; i < period; ++i) {
            if (ans[i + md] != ans[i + period + md]) {
                gt = false;
                break;
            }
        }
        if (gt)r = md;
        else l = md;
    }
    while (q--) {
        ll x; cin >> x;
        if (x >= ans.size()) {
            x = (x - r) % period + r;
        }
        cout << ans[x] + 1 << " ";
    }
}
//        : mandei
/*
1 2 1 1
1 1 1 2
1 1 2 1*/
```

Task F

```
//#pragma GCC optimize("O3","unroll-loops")
#include<iostream>
#include<vector>
#include<queue>
#include<bitset>
#include<set>
#include<iomanip>
#include<random>
#include<algorithm>
#include<map>
#include<cmath>
using namespace std;
```

```
using ll = long long;
using ld = long double;
const int mod = 1e9 + 7;
const int INF = 1e9 + 7;
const ll LINF = 1e18 + 7;
const ld LDINF = 1e8;
mt19937 rnd;
const int maxn = 1000000;
vector<int>a, cur_ind;
ll bl = 0;
vector<vector<pair<int, ll>>>g;
vector<int>mt;
bool cmp(pair<ll, ll>a, pair<ll, ll>b) {
    return a.second > b.second;
}
bool dfs(int v) {
    //cout << v << "\n";
    for (auto [u, w] : g[v]) {
        if (mt[u] == -1) {
            //cout << v << " is capped with " << u << "\n";
            mt[u] = v;
            return true;
        }
    }
    for (auto [u, w] : g[v]) {
        if (mt[u]!=v and dfs(mt[u])) {
            //cout << v << " is capped with " << u << "\n";
            mt[u] = v;
            return true;
        }
    }
    return false;
}
int main() {
    cin.tie()->sync_with_stdio(0);
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout
    );
    int h, w; cin >> h >> w;
    vector<vector<ll>>a(h, vector<ll>(w));
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            char c; cin >> c;
            a[i][j] = c - '0';
        }
    }
    g.resize(h * w);
    mt.resize(h * w, -1);
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            g[i * w + j].push_back({ ((i + 1) % h) * w + j,a[i][j] * a[(i
     + 1) % h][j] });
            g[i * w + j].push_back({ ((i - 1 + h) % h) * w + j, a[i][j] *
```

```
 a[(i + h - 1) % h][j] });
            g[i * w + j].push_back({ i * w + (j + 1) % w, a[i][j] * a[i
][(j + 1) % w] });
            g[i * w + j].push_back({ i * w + (j - 1 + w) % w, a[i][j] * a
[i][(j + w - 1) % w] });
            sort(g[i * w + j].begin(), g[i * w + j].end(), cmp);
        }
    }
    /*for (auto x : g[34])cout << x.first << " " << x.second << "  ";
    cout << "\n";
    return 0;*/
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            if ((i + j) % 2 == 0)dfs(i * w + j);
        }
    }
    ll ans = 0;
    vector<vector<char>>kw(h, vector<char>(w,'.'));
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            if ((i + j) % 2 == 1) {
                int I = mt[i * w + j] / w, J = mt[i * w + j] % w;
                ans += a[i][j] * a[I][J];
                if (I == i) {
                    kw[I][min(j, J)] = 'L';
                    kw[I][max(j, J)] = 'R';
                }
                else {
                    kw[min(i, I)][J] = 'T';
                    kw[max(i, I)][J] = 'B';
                }
            }
        }
    }
    cout << ans << "\n";
    for (auto c : kw) {
        for (auto x : c)cout << x;
        cout << "\n";
    }
}
//        : mandei
```