

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	89	0	429

Task A ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

signed main() {
#ifdef ONPC
    freopen("in.txt", "r", stdin);
#endif // ONPC
    ios_base::sync_with_stdio(0), cin.tie(0);
    int t = 1;
    //cin >> t;
    while (t--) solve();
    return 0;
}

void solve() {
    ll n;
    cin >> n;
    cout << n - 1;
}
//find_by_order
//order_of_key
```

Task B ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

signed main() {
#ifdef ONPC
    //freopen("in.txt", "r", stdin);
#endif // ONPC
    ios_base::sync_with_stdio(0), cin.tie(0);
    int t = 1;
    //cin >> t;
    while (t--) solve();
    return 0;
}

ld dist(pair<ld, ld> f, pair<ld, ld> s) {
    return (f.X - s.X) * (f.X - s.X) + (f.Y - s.Y) * (f.Y - s.Y);
}

void solve() {
    cout << fixed << setprecision(6);
    int n;
    cin >> n;
    vector<pair<ld, ld>> v;
    for (int i = 0; i < n; ++i) {
        ld x, y;
        cin >> x >> y;
        v.pb({x, y});
    }
    if (n == 6) {
        pair<ld, ld> start = {1001, 1001};
        int ind = 0;
        for (int i = 0; i < n; ++i) {
            if (v[i].Y < start.Y || (v[i].Y == start.Y && v[i].X < start.X)) {
                ind = i;
                start.X = v[i].X;
                start.Y = v[i].Y;
            }
        }
        pair<ld, ld> nxt = {5005, 5005};
        int ind2 = 0;
        cout << start.X << ' ' << start.Y << '\n';
        for (int i = 0; i < n; ++i) {
            if (ind == i) continue;
            if (dist(nxt, start) > dist(v[i], start)) {
                nxt = v[i];
                ind2 = i;
            }
        }
        cout << nxt.X << ' ' << nxt.Y << '\n';
        pair<ld, ld> lol = {5005, 5005};
        for (int i = 0; i < n; ++i) {
            if (ind == i || ind2 == i) continue;
            if (dist(nxt, lol) > dist(nxt, v[i])) {
                lol = v[i];
            }
        }
        cout << lol.X << ' ' << lol.Y << '\n';
        cout << endl;
        return;
    }
    pair<ld, ld> vec1 = {v[1].X - v[0].X, v[1].Y - v[0].Y};
```

```

pair<ld, ld> vec2 = {v[2].X - v[1].X, v[2].Y - v[1].Y};
pair<ld, ld> dv1 = {-vec1.X, -vec1.Y};
pair<ld, ld> dv2 = {-vec2.X, -vec2.Y};
vector<pair<ld, ld>> ans;
ans = v;
ans.pb({v[0].X + (ld)2 * vec2.X, v[0].Y + (ld)2 * vec2.Y});
ans.pb({ans.back().X + dv1.X, ans.back().Y + dv1.Y});
ans.pb({ans.back().X + dv2.X, ans.back().Y + dv2.Y});
for (auto d : ans) {
    cout << d.X << ' ' << d.Y << '\n';
}
cout << endl;
}
//find_by_order
//order_of_key

```

Task C ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

const int INF = 1e9 + 7;
const int N = 1e4 + 5;
string s;
int ans, dp[N][505], len;
vector<int> g[30];

signed main() {
#ifdef ONPC
    freopen("in.txt", "r", stdin);
#endif // ONPC
    ios_base::sync_with_stdio(0), cin.tie(0);
    cin >> s;
    len = sz(s);
    s = '#' + s;
    for (int i = 1; i <= len; ++i) {
        g[s[i] - 'a'].pb(i);
    }
    int t = 1;
    cin >> t;
    while (t--) solve();
    cout << ans;
    return 0;
}

void solve() {
    string a;
    cin >> a;
    int cur = len;
    int n = sz(a);
    a = '#' + a;
    for (int i = 0; i <= n; ++i) {
        for (int j = 0; j <= len; ++j) {
            dp[i][j] = INF;
        }
        dp[i][0] = 0;
    }
    for (int i = 1; i <= n; ++i) {
        if (i == 1) {
            for (auto j : g[a[i] - 'a']) {
                dp[i][j] = j - 1;
                cur = min(cur, dp[i][j] + len - j);
                //cout << i << ' ' << j << ' ' << dp[i][j] << '\n';
            }
            continue;
        }
        for (auto j : g[a[i] - 'a']) {
            dp[i][j] = j - 1;
            for (int z = 0; z < sz(g[a[i - 1] - 'a']); ++z) {
                auto d = g[a[i - 1] - 'a'][z];
                if (d >= j) break;
                dp[i][j] = min(dp[i][j], dp[i - 1][d] + j - d - 1);
            }
            //cout << i << ' ' << j << ' ' << dp[i][j] << '\n';
            cur = min(cur, dp[i][j] + len - j);
        }
    }
    //cout << cur << '\n';
    ans += cur;
}
```

```
}  
//find_by_order  
//order_of_key
```

Task D ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
//using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

signed main() {
#ifdef ONPC
    freopen("in.txt", "r", stdin);
#endif // ONPC
    ios_base::sync_with_stdio(0), cin.tie(0);
    int t = 1;
    //cin >> t;
    while (t--) solve();
    return 0;
}

const int N = 1005;
int n, m, cc[N][N], uk, dist[N * N];
pair<int, int> a[N][N], s, f;
pair<int, int> g[N][N];
vector<pair<int, int>> ng[N * N], gr[N][N];
bool used[N][N];
vector<pair<int, int>> ord;
vector<pair<int, int>> vec;
vector<vector<pair<int, int>>> comp;

void dfs(int x, int y) {
    used[x][y] = 1;
    auto d = g[x][y];
    if (d.X != 0) {
        if (!used[d.X][d.Y]) {
            dfs(d.X, d.Y);
        }
    }
    ord.pb({x, y});
}

void dfs2(int x, int y) {
    cc[x][y] = uk;
    vec.pb({x, y});
    used[x][y] = 1;
    for (auto d : gr[x][y]) {
        if (!used[d.X][d.Y]) {
            dfs2(d.X, d.Y);
        }
    }
}

void solve() {
    cin >> n >> m;
    cin >> s.X >> s.Y >> f.X >> f.Y;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            cin >> a[i][j].X >> a[i][j].Y;
            pair<int, int> nw = {a[i][j].X + i, a[i][j].Y + j};
            if (nw.X > 0 && nw.X <= n && nw.Y > 0 && nw.Y <= m) {
                g[i][j] = nw;
                gr[nw.X][nw.Y].pb({i, j});
            }
        }
    }
    for (int i = 1; i <= n; ++i) {
```

```

        for (int j = 1; j <= m; ++j) {
            if (!used[i][j]) {
                dfs(i, j);
            }
        }
    }
    reverse(ord.begin(), ord.end());
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            used[i][j] = 0;
        }
    }
    for (auto d : ord) {
        if (!used[d.X][d.Y]) {
            vec.clear();
            dfs2(d.X, d.Y);
            ++uk;
            comp.pb(vec);
        }
    }
    int start = cc[s.X][s.Y], finish = cc[f.X][f.Y];
    if (start == finish) {
        cout << 0;
        return;
    }
    for (int i = 0; i < uk; ++i) {
        for (int j = 0; j < uk; ++j) {
            if (i == j) continue;
            int dist = 1e9;
            for (auto d : comp[i]) {
                for (auto dd : comp[j]) {
                    int curx = dd.X - d.X;
                    int cury = dd.Y - d.Y;
                    curx -= a[d.X][d.Y].X;
                    cury -= a[d.X][d.Y].Y;
                    dist = min(dist, abs(curx) + abs(cury));
                }
            }
            ng[i].pb({j, dist});
        }
        dist[i] = 1e9 + 7;
    }
    priority_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> Q;
    dist[start] = 0;
    Q.push({0, start});
    while (!Q.empty()) {
        auto t = Q.top();
        Q.pop();
        if (t.X != dist[t.Y]) {
            continue;
        }
        for (auto d : ng[t.Y]) {
            if (dist[d.X] > t.X + d.Y) {
                dist[d.X] = t.X + d.Y;
                Q.push({dist[d.X], d.X});
            }
        }
    }
    cout << dist[finish];
}
//find_by_order
//order_of_key

```

Task E ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

signed main() {
#ifdef ONPC
    //freopen("in.txt", "r", stdin);
#endif // ONPC
    //ios_base::sync_with_stdio(0), cin.tie(0);
    int t = 1;
    //cin >> t;
    while (t--) solve();
    return 0;
}

int n, m, B;
vector<pair<int, int>> v;
set<int> Q[16];
set<int> cant;

void solve() {
    cin >> n >> m >> B;
    for (int i = 0; i < B; ++i) {
        int x, y;
        cin >> x >> y;
        --x, --y;
        v.pb({x, y});
    }
    int need = (1 << (B - 1));
    int y = 1;
    int iter = 0;
    for (int i = 1; i <= need; ++i) {
        while (cant.count(y)) {
            y += m;
        }
        cout << "?_ " << 1 + v[0].X << '␣' << y + v[0].Y << '␣';
        Q[1].insert(y);
        y += m;
        ++iter;
        while (cant.count(y)) {
            y += m;
        }
        cout << 1 + v[0].X << '␣' << y + v[0].Y << endl;
        Q[1].insert(y);
        y += m;
        int xx, yy;
        cin >> xx >> yy;
        int cur = 0;
        if (1 <= xx && xx <= n) {
            for (auto d : Q[1]) {
                if (d <= yy && d + m > yy) {
                    cur = d;
                    break;
                }
            }
            if (cur != 0)
                Q[1].erase(cur);
            else {
                cant.insert(((yy - 1) / m + 1) * m);
            }
        }
    }
}
```



```

for (int i = 1; i < B; ++i) {
    while (!Q[i].empty() && (sz(Q[i + 1]) < (1 << (B - i - 1))) && (B - i - 2 < 0 || (sz(Q[i + 2]) < (1 << (B - i - 2))))) {
        auto t = *(Q[i].begin());
        cout << "?_ " << 1 + v[i].X << '_ ' << t + v[i].Y << '_';
        ++iter;
        int nxt = 0;
        Q[i].erase(Q[i].begin());
        int ind = i;
        if (Q[i].empty()) {
            nxt = *(Q[i + 1].begin());
            Q[i + 1].erase(Q[i + 1].begin());
            ind = i + 1;
        } else {
            nxt = *(Q[i].begin());
            Q[i].erase(Q[i].begin());
        }
        cout << 1 + v[ind].X << '_ ' << nxt + v[ind].Y << endl;
        //assert(iter < 8191);
        Q[i + 1].insert(t);
        Q[ind + 1].insert(nxt);
        int xx, yy;
        cin >> xx >> yy;
        int cur = 0;
        if (1 <= xx && xx <= n) {
            for (auto d : Q[i]) {
                if (d <= yy && d + m > yy) {
                    cur = d;
                    break;
                }
            }
            if (cur != 0)
                Q[i].erase(cur);
            for (auto d : Q[i + 1]) {
                if (d <= yy && d + m > yy) {
                    cur = d;
                    break;
                }
            }
            if (cur != 0)
                Q[i + 1].erase(cur);
            for (auto d : Q[i + 2]) {
                if (d <= yy && d + m > yy) {
                    cur = d;
                    break;
                }
            }
            if (cur != 0)
                Q[i + 2].erase(cur);
        }
        if (sz(Q[B])) {
            break;
        }
    }
    if (sz(Q[B])) {
        break;
    }
}
cout << "!_ ";
assert(!Q[B].empty());
auto t = *(Q[B].begin());
cout << 1 << '_ ' << t << endl;
}
//find_by_order
//order_of_key

```

Task F ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
#define X first
#define Y second
#define pb push_back
#define sz(a) (int)a.size()
using ord = tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>;

void solve();

signed main() {
#ifdef ONPC
    freopen("in.txt", "r", stdin);
#endif // ONPC
    ios_base::sync_with_stdio(0), cin.tie(0);
    int t = 1;
    //cin >> t;
    while (t--) solve();
    return 0;
}

int n, m;
const ll mod = 1e9 + 7;
ll fact[28];

ll binp(ll d, ll st) {
    if (st == 0) return 1;
    if (st == 1) return d;
    ll gg = binp(d, st / 2);
    if (st & 1) return gg * gg * d;
    return gg * gg;
}

void solve() {
    cin >> n >> m;
    if (n == 2) {
        cout << 1;
        return;
    }
    fact[1] = 1;
    for (int i = 2; i <= n; ++i) {
        fact[i] = fact[i - 1] * i;
    }
    m = (n * n * n - n) / 6;
    ll zer = n - 1 + (n - 1) * (n - 2);
    for (int i = 1; i < zer; ++i) cout << 0 << '␣';
    ll all = binp(n, n - 2);
    //cout << all * 2 << ' ' << binp(3, zer) - 1 << '\n';
    ll b1 = n;
    for (int i = zer; i <= m; ++i) {
        cout << b1 << '␣';
        b1 *= fact[n - 1] / 2;
    }
}

//find_by_order
//order_of_key
```