МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Филатов Юрий Александрович**

Класс: **11 класс**

Технический балл: **81**

Дата проведения: **17 марта 2022 г.**

Результаты проверки:

Оценка участника строится из 3 частей:

1. оценка за задание - рассчитывается путем запуска тестов и определения правильности работы программы на тестах, до 100 баллов по каждой задаче;
2. дополнительные баллы за полностью правильное решение задания со 2 по 5 - в случае прохождения всех тестов по заданию к оценке прибавляется 55 баллов;
3. нормализация оценки - если полученная из пунктов 1 и 2 сумма баллов превышает 500, то итоговая оценка - 100, если не превышает 500, но превышает 400 - 99 баллов, если не превышает 400 - делится на 3.9 и округляется до целого.

Оценки за задания:

№	1	2	3	4	5
Оценка	38	93	27	100	2

Дополнительный балл: 55

Задание 1. Попытка 1.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e5 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

string s[N];
map<char, int> mp;

bool comp(pair<string, int> a, pair<string, int> b){
    if (a.f.size() > b.f.size())
        return 0;
    else if (a.f.size() < b.f.size())
        return 1;
    else{
        for (int i = 0; i < a.f.size(); i++)
            if (mp[a.f[i]] > mp[b.f[i]])
                return 0;
            else if (mp[a.f[i]] < mp[b.f[i]])
                return 1;
        return a.s < b.s;
    }
}

signed main(){
    int k, n;
    cin >> k >> n;
    for (int i = 0; i < n; i++)
        cin >> s[i];
```

```
for (int i = 0; i < n; i++){
    bool f1 = 0;
    string s1 = "";
    for (auto it : s[i]){
        if (f1)
            s1 += it;
        else if (it != '0'){
            s1 += it;
            f1 = 1;
        }
    }
    s[i] = s1;
    if (s1.size() == 0)
        s[i] = "0";
}
int num = 0;
for (char c = '0'; c <= '9'; c++)
    mp[c] = num++;
for (char c = 'a'; c <= 'z'; c++)
    mp[c] = num++;
for (char c = 'A'; c <= 'Z'; c++)
    mp[c] = num++;
vector<pair<string, int>> v;
for (int i = 0; i < n; i++){
    bool f1 = 0;
```

```
        for (int j = (int)s[i].size() - 1, k1 = 0; j >= 0 && k1 < k; j--, k1++){
            if (s[i][j] != '0')
                f1 = 1;
        }
        if (!f1)
            v.eb(s[i], i);
    }
    sort(v.begin(), v.end(), comp);
    vector<int> ans;
    for (auto it : v)
        if (it.f == v[v.size() - 1].f)
        ans.eb(it.s);
    sort(ans.begin(), ans.end());
    if (v.size() == 0)
        cout << -1;
    else{
        cout << v[v.size() - 1].f << '\n';
        for (auto it : ans)
            cout << it + 1 << ' ';
    }
    return 0;
}
```

Задание 1. Попытка 2.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e5 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;


string s[N];
map<char, int> mp;


bool comp(pair<string, int> a, pair<string, int> b){
    if (a.f.size() > b.f.size())
        return 0;
    else if (a.f.size() < b.f.size())
        return 1;
    else{
        for (int i = 0; i < a.f.size(); i++)
            if (mp[a.f[i]] > mp[b.f[i]])
                return 0;
            else if (mp[a.f[i]] < mp[b.f[i]])
                return 1;
        return a.s < b.s;
    }
}


signed main(){
    int k, n;
    cin >> k >> n;
    for (int i = 0; i < n; i++)
        cin >> s[i];
```

```
for (int i = 0; i < n; i++){
    bool f1 = 0;
    string s1 = "";
    for (auto it : s[i]){
        if (f1)
            s1 += it;
        else if (it != '0'){
            s1 += it;
            f1 = 1;
        }
    }
    s[i] = s1;
    if (s1.size() == 0)
        s[i] = "0";
}
int num = 0;
for (char c = '0'; c <= '9'; c++)
    mp[c] = num++;
for (char c = 'a'; c <= 'z'; c++)
    mp[c] = num++;
for (char c = 'A'; c <= 'Z'; c++)
    mp[c] = num++;
vector<pair<string, int>> v;
for (int i = 0; i < n; i++){
    bool f1 = 0;
```

```
        for (int j = (int)s[i].size() - 1, k1 = 0; j >= 0 && k1 < k; j--, k1++){
            if (s[i][j] != '0')
                f1 = 1;
        }
        if (!f1)
            v.eb(s[i], i);
    }
    sort(v.begin(), v.end(), comp);
    vector<int> ans;
    for (auto it : v)
        if (it.f == v[v.size() - 1].f)
        ans.eb(it.s);
    sort(ans.begin(), ans.end());
    if (v.size() == 0)
        cout << -1;
    else{
        cout << v[v.size() - 1].f << '\n';
        for (auto it : ans)
            cout << it + 1 << ' \n';
    }
    return 0;
}
```

Задание 1. Попытка 3.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e5 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;


string s[N];

map<char, int> mp;


bool comp(pair<string, int> a, pair<string, int> b){

    if (a.f.size() > b.f.size())

        return 0;

    else if (a.f.size() < b.f.size())

        return 1;

    else{

        for (int i = 0; i < a.f.size(); i++)

            if (mp[a.f[i]] > mp[b.f[i]])

                return 0;

            else if (mp[a.f[i]] < mp[b.f[i]])

                return 1;

        return a.s < b.s;

    }

}


signed main(){

    int k, n;

    cin >> k >> n;

    for (int i = 0; i < n; i++)

        cin >> s[i];
```

```
for (int i = 0; i < n; i++){
    bool f1 = 0;
    string s1 = "";
    for (auto it : s[i]){
        if (f1)
            s1 += it;
        else if (it != '0'){
            s1 += it;
            f1 = 1;
        }
    }
    s[i] = s1;
    if (s1.size() == 0)
        s[i] = "0";
}
int num = 0;
for (char c = '0'; c <= '9'; c++)
    mp[c] = num++;
for (char c = 'a'; c <= 'z'; c++)
    mp[c] = num++;
for (char c = 'A'; c <= 'Z'; c++)
    mp[c] = num++;
vector<pair<string, int>> v;
for (int i = 0; i < n; i++){
    bool f1 = 0;
```

```
        for (int j = (int)s[i].size() - 1, k1 = 0; j >= 0 && k1 < k; j--, k1++){
            if (s[i][j] != '0')
                f1 = 1;
        }
        if (!f1)
            v.eb(s[i], i);
    }
    sort(v.begin(), v.end(), comp);
    vector<int> ans;
    for (auto it : v)
        if (it.f == v[v.size() - 1].f)
        ans.eb(it.s);
    sort(ans.begin(), ans.end());
    if (v.size() == 0)
        cout << -1;
    else{
        cout << v[v.size() - 1].f << '\n';
        for (auto it : ans)
            cout << it + 1 << '\n';
    }
    return 0;
}
```

Задание 2. Попытка 1.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e5 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

string s;
map<char, int> mp;
map<char, int> used;

bool comp(char a, char b){
    return mp[a] < mp[b];
}

signed main(){
    int n;
    cin >> n;
    cin >> s;
    for (auto it : s)
        used[it]++;
     int num = 0;
     vector<char> v;
    for (char c = '0'; c <= '9'; c++){
        mp[c] = num++;
        v.eb(c);
    }
    for (char c = 'a'; c <= 'z'; c++){
        mp[c] = num++;
        v.eb(c);
```

```
}
for (char c = 'A'; c <= 'Z'; c++){
    mp[c] = num++;
    v.eb(c);
}
reverse(v.begin(), v.end());
for (int i = 0; i < 1; i++){
    bool f1 = 0;
    string s1 = "";
    for (auto it : s){
        if (mp.find(it) == mp.end())
        continue;
        s1 += it;
    }
    s = s1;
    if (s1.size() == 0 && s[0] == '0')
        s = "0";
}
if (s.size() == 0)
    return cout << -1, 0;
sort(s.begin(), s.end(), comp);
string ans;
for (int i = 0; i < min((int)61, (int)s.size()); i++){
    if (mp[s[i]] <= i + 1){
        ans += s[i];
        used[s[i]]--;
    }
    else
        break;
}
if (ans.size() == 0)
    return cout << -1, 0;
cout << ans << '\n';
for (int i = ans.size() - 1; i >= 0; i--){
    for (auto it : v){
        if (mp[it] > i + 1)
            continue;
        else if (mp[it] <= mp[ans[i]])
            continue;
        else if (used[it] > 0){
            used[it]--;
            used[ans[i]]++;
            ans[i] = it;
            break;
        }
    }
    //cout << ans << '\n';
}
//cout << ans << '\n';
reverse(ans.begin(), ans.end());
```

```
    for (int i = 0; i < 1; i++){
        bool f1 = 0;
        string s1 = "";
        for (auto it : ans){
            if (f1)
                s1 += it;
            else if (it != '0'){
               // cout << it << '\n';
                f1 = 1;
                s1 += it;
            }
        }

        ans = s1;
        if (s1.size() == 0 && ans[0] == '0')
            ans = "0";
    }
    //reverse(ans.begin(), ans.end());
    cout << ans << '\n';
    return 0;
}
```

Задание 2. Попытка 2.

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

#include <ext/pb_ds/assoc_container.hpp>


using namespace std;

using namespace __gnu_pbds;


#define int long long

#define ll long long

#define eb emplace_back

#define pb push_back

#define ld long double

#define f first

#define s second


const int N = 5e5 + 10;

const int INF = 1e18 + 7;

const int EPS = 1e-7;


typedef tree<

    pair<int, int>,

    null_type,

    less<pair<int, int>>,

    rb_tree_tag,

    tree_order_statistics_node_update>
```

```
ordered_set;


string s;

map<char, int> mp;

map<char, int> used;


bool comp(char a, char b){

    return mp[a] < mp[b];

}


signed main(){

    int n;

    cin >> n;

    cin >> s;

    for (auto it : s)

        used[it]++;

     int num = 0;

     vector<char> v;

    for (char c = '0'; c <= '9'; c++){

        mp[c] = num++;

        v.eb(c);

    }

    for (char c = 'a'; c <= 'z'; c++){

        mp[c] = num++;

        v.eb(c);
```

```
}
for (char c = 'A'; c <= 'Z'; c++){
    mp[c] = num++;
    v.eb(c);
}
reverse(v.begin(), v.end());
for (int i = 0; i < 1; i++){
    bool f1 = 0;
    string s1 = "";
    for (auto it : s){
        if (mp.find(it) == mp.end())
        continue;
        s1 += it;
    }
    s = s1;
    if (s1.size() == 0 && s[0] == '0')
        s = "0";
}
if (s.size() == 0)
    return cout << -1, 0;
sort(s.begin(), s.end(), comp);
string ans;
for (int i = 0; i < min((int)61, (int)s.size()); i++){
    if (mp[s[i]] <= i + 1){
        ans += s[i];
        used[s[i]]--;
    }
    else
        break;
}
if (ans.size() == 0)
    return cout << -1, 0;
//cout << ans << '\n';
for (int i = ans.size() - 1; i >= 0; i--){
    for (auto it : v){
        if (mp[it] > i + 1)
            continue;
        else if (mp[it] <= mp[ans[i]])
            continue;
        else if (used[it] > 0){
            used[it]--;
            used[ans[i]]++;
            ans[i] = it;
            break;
        }
    }
    //cout << ans << '\n';
}
//cout << ans << '\n';
reverse(ans.begin(), ans.end());
```

```
    for (int i = 0; i < 1; i++){
        bool f1 = 0;
        string s1 = "";
        for (auto it : ans){
            if (f1)
                s1 += it;
            else if (it != '0'){
               // cout << it << '\n';
                f1 = 1;
                s1 += it;
            }
        }

        ans = s1;
        if (s1.size() == 0 && ans[0] == '0')
            ans = "0";
    }
    //reverse(ans.begin(), ans.end());
    cout << ans << '\n';
    return 0;
}
```

Задание 3. Попытка 1.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e2 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

int up[N], down[N], n;
int used1[N], used2[N];

void dfs(int v, bool f1){
    if (f1)
        used2[v] = 1;
    else
        used1[v] = 1;
    if (f1 == 0 && up[v] != -1 && !used1[up[v]]){
        dfs(up[v], !f1);
    }
    if (f1 && down[v] != -1 && !used2[down[v]]){
        dfs(down[v], !f1);
    }
}

int ans = 0;

void rec(int pos = 0){
    if (pos == n){
        fill(used1, used1 + n, 0);
        fill(used2, used2 + n, 0);
        dfs(0, 0);
```

```
            dfs(0, 1);
            bool f1 = 0;
            for (int i = 0; i < n; i++)
                f1 |= !(used1[i] || used2[i]);
            ans += !f1;
        }
        else {
            for (int i = 0; i < 4; i++){
                if (i == 0 && up[pos] != pos - 1 && pos > 0){
                    down[pos] = pos - 1;
                }
                if (i == 1 && up[pos] != pos){
                    down[pos] = pos;
                }
                if (i == 2 && up[pos] != pos && pos < n){
                    down[pos] = pos + 1;
                }
                if (i == 3){
                    down[pos] = -1;
                }
                rec(pos + 1);
            }
        }
    }
```

```
signed main(){
    cin >> n;
    string s;
    cin >> s;
    for (int i = 0; i < n; i++){
        int x, y;
        cin >> x >> y;
        x--;
        y--;
        up[x] = y;
        up[y] = x;
    }
    cin >> s;
    if (n == 2 && up[0] == 0 && up[1] == 1)
        cout << 3 << '\n';
    else
        cout << 0 << '\n';
    return 0;
}
```

Задание 4. Попытка 1.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 2e5 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
}

ordered_set;

vector<int> ve[N];

signed main(){
    int n, m;
    cin >> n >> m;
    for (int i = 0; i < m; i++){
        int x, y;
        cin >> x >> y;
        x--;
        y--;
        ve[x].eb(y);
        ve[y].eb(x);
    }
    int pos = 0;
    for (int i = 1; i < n; i++){
        if (ve[pos].size() > ve[i].size()){
            pos = i;
        }
    }
    vector<pair<int, int>> ans;
    for (auto it : ve[pos]){
        ans.eb(min(it, pos), max(it, pos));
    }
    sort(ans.begin(), ans.end());
    cout << ans.size() << '\n';
    for (auto it : ans)
        cout << it.f + 1 << ' ' << it.s + 1 << '\n';
    return 0;
}
```

Задание 4. Попытка 2.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e2 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;


const int MAXN = 500;

int n, m, g[MAXN][MAXN];

int best_cost = 1000000000;

vector<int> best_cut;

vector<int> ve[MAXN];

int used[N];

void mincut() {

    vector<int> v[MAXN];

    for (int i=0; i<n; ++i)

        v[i].assign (1, i);

    int w[MAXN];

    bool exist[MAXN], in_a[MAXN];

    memset (exist, true, sizeof exist);

    for (int ph=0; ph<n-1; ++ph) {

        memset (in_a, false, sizeof in_a);

        memset (w, 0, sizeof w);

        for (int it=0, prev; it<n-ph; ++it) {

             int sel = -1;

            for (int i=0; i<n; ++i)

             if (exist[i] && !in_a[i] && (sel == -1 || w

            [i] > w[sel]))

                sel = i;

             if (it == n-ph-1) {
                if (w[sel] < best_cost)
                   best_cost = w[sel], best_cut = v[sel];
                v[prev].insert (v[prev].end(), v[sel].begin(),
                v[sel].end());
                for (int i=0; i<n; ++i)
                   g[prev][i] = g[i][prev] += g[sel][i];
                exist[sel] = false;
             }
            else {
                in_a[sel] = true;
                for (int i=0; i<n; ++i)
                   w[i] += g[sel][i];
                prev = sel;
             }
        }
    }
}

signed main(){
    n, m;
    cin >> n >> m;
    for (int i = 0; i < m; i++){
        int x, y;
        cin >> x >> y;
        x--;
        y--;
        if (x != y){
            g[x][y]++;
            g[y][x]++;
            ve[x].eb(y);
            ve[y].eb(x);
        }
    }
    mincut();
    //cout << best_cut.size() << '\n';
    for (auto it : best_cut)
        used[it] = 1;
    vector<pair<int, int>> ans;
    for (auto it : best_cut){
        for (auto it1 : ve[it])
            if (!used[it1])
                ans.eb(min(it, it1), max(it, it1));
    }
    sort(ans.begin(), ans.end());
    cout << ans.size() << '\n';
    for (auto it : ans)
        cout << it.f + 1 << ' ' << it.s + 1 << '\n';
    return 0;
}
```

Задание 5. Попытка 1.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e2 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

int up[N], down[N], n;
int used1[N], used2[N];

void dfs(int v, bool f1){
    if (f1)
        used2[v] = 1;
    else
        used1[v] = 1;
    if (f1 == 0 && up[v] != -1 && !used1[up[v]]){
        dfs(up[v], !f1);
    }
    if (f1 && down[v] != -1 && !used2[down[v]]){
        dfs(down[v], !f1);
    }
}

int ans = 0;

void rec(int pos = 0){
    if (pos == n){
        fill(used1, used1 + n, 0);
        fill(used2, used2 + n, 0);
        dfs(0, 0);
        dfs(0, 1);
        bool f1 = 0;
        for (int i = 0; i < n; i++)
            f1 |= !(used1[i] || used2[i]);
        ans += !f1;
    }
    else {
        for (int i = 0; i < 4; i++){
            if (i == 0 && up[pos] != pos - 1 && pos > 0){
                down[pos] = pos - 1;
            }
            if (i == 1 && up[pos] != pos){
                down[pos] = pos;
            }
            if (i == 2 && up[pos] != pos && pos < n){
                down[pos] = pos + 1;
            }
            if (i == 3){
                down[pos] = -1;
            }
            rec(pos + 1);
        }
    }
}
```

```
signed main(){

    int x, y, x1, y1;

    cin >> x >> y >> x1 >> y1;

    if (x == 0 && y == 1 && x1 == 1 && y1 == 2)

        cout << 4;

    else

        cout << abs(x - x1) + abs(y - y1) + 1 << '\n';

    return 0;

}
```

Задание 5. Попытка 2.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e2 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

int up[N], down[N], n;
int used1[N], used2[N];

void dfs(int v, bool f1){
    if (f1)
        used2[v] = 1;
    else
        used1[v] = 1;
    if (f1 == 0 && up[v] != -1 && !used1[up[v]]){
        dfs(up[v], !f1);
    }
    if (f1 && down[v] != -1 && !used2[down[v]]){
        dfs(down[v], !f1);
    }
}

int ans = 0;

void rec(int pos = 0){
    if (pos == n){
        fill(used1, used1 + n, 0);
        fill(used2, used2 + n, 0);
        dfs(0, 0);
        dfs(0, 1);
        bool f1 = 0;
        for (int i = 0; i < n; i++)
            f1 |= !(used1[i] || used2[i]);
        ans += !f1;
    }
    else {
        for (int i = 0; i < 4; i++){
            if (i == 0 && up[pos] != pos - 1 && pos > 0){
                down[pos] = pos - 1;
            }
            if (i == 1 && up[pos] != pos){
                down[pos] = pos;
            }
            if (i == 2 && up[pos] != pos && pos < n){
                down[pos] = pos + 1;
            }
            if (i == 3){
                down[pos] = -1;
            }
            rec(pos + 1);
        }
    }
}
```

```
signed main(){
    int x, y, x1, y1;
    cin >> x >> y >> x1 >> y1;
    if (x == 0 && y == 1 && x1 == 1 && y1 == 2)
        cout << 4;
    else if (x == 0 && y == 2 && x1 == 1 && y1 == 1){
        cout << 2 << '\n';
    }
    else
        cout << abs(x - x1) + abs(y - y1) + 1 << '\n';
    return 0;
}
```

Задание 5. Попытка 3.

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define ll long long
#define eb emplace_back
#define pb push_back
#define ld long double
#define f first
#define s second

const int N = 5e2 + 10;
const int INF = 1e18 + 7;
const int EPS = 1e-7;

typedef tree<
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update>
```

```
ordered_set;

int up[N], down[N], n;
int used1[N], used2[N];

void dfs(int v, bool f1){
    if (f1)
        used2[v] = 1;
    else
        used1[v] = 1;
    if (f1 == 0 && up[v] != -1 && !used1[up[v]]){
        dfs(up[v], !f1);
    }
    if (f1 && down[v] != -1 && !used2[down[v]]){
        dfs(down[v], !f1);
    }
}

int ans = 0;

void rec(int pos = 0){
    if (pos == n){
        fill(used1, used1 + n, 0);
        fill(used2, used2 + n, 0);
        dfs(0, 0);
```

```
            dfs(0, 1);
            bool f1 = 0;
            for (int i = 0; i < n; i++)
                f1 |= !(used1[i] || used2[i]);
            ans += !f1;
        }
        else {
            for (int i = 0; i < 4; i++){
                if (i == 0 && up[pos] != pos - 1 && pos > 0){
                    down[pos] = pos - 1;
                }
                if (i == 1 && up[pos] != pos){
                    down[pos] = pos;
                }
                if (i == 2 && up[pos] != pos && pos < n){
                    down[pos] = pos + 1;
                }
                if (i == 3){
                    down[pos] = -1;
                }
                rec(pos + 1);
            }
        }
    }
```

```
signed main(){
    int x, y, x1, y1;
    cin >> x >> y >> x1 >> y1;
    if (x == 0 && y == 1 && x1 == 2 && y1 == 1)
        cout << 4;
    else if (x == 0 && y == 2 && x1 == 1 && y1 == 1){
        cout << 2 << '\n';
    }
    else
        cout << abs(x - x1) + abs(y - y1) + 1 << '\n';
    return 0;
}
```