

Олимпиада СПбГУ по информатике 2018/19 учебного года

Туров Кирилл Васильевич

A	B	C	D	E	F	Sum
100	100	100	60	21	0	381

Task A (100)

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

using namespace std;
using namespace __gnu_cxx;

int main()
{
    //rope<int> k;
    int n, m;
    cin >> n >> m;
    for (n; n <= m; n *= 2) {
        if (n == m) {
            cout << "Yes";
            return 0;
        }
    }
    cout << "No";
    return 0;
}
```

Task B (100)

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

using namespace std;

int main()
{
    int n;
    cin >> n;
    string s;
    cin >> s;
    for (int i = 0; i + 1 < s.size(); i++) {
        if (s[i] == 'o' && s[i + 1] == 'r' ||
            s[i] == 'r' && s[i + 1] == 'o') {
            cout << "Yes";
            return 0;
        }
    }
    for (int i = 0; i + 2 < s.size(); i++) {
        if (s[i] == 'o' && s[i + 2] == 'r') {
            cout << "Yes";
            return 0;
        }
    }
    cout << "No";
    return 0;
}
```

Task C (100)

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

using namespace std;

int n;
vector<vector<int>> > g;
vector<int> sz;
vector<int> ans;

void dfs(int v, int p = -1) {
    sz[v] = 1;
    int maxx = -1;
    for (int u : g[v]) {
        if (u != p) {
            dfs(u, v);
            sz[v] += sz[u];
            maxx = max(maxx, sz[u]);
        }
    }
    maxx = max(maxx, n - sz[v]);
    ans[v] = maxx + 1;
}

int main()
{
    cin >> n;
    g.resize(n);
    sz.resize(n);
    ans.resize(n);
    for (int i = 0; i < n - 1; i++) {
        int v, u;
        cin >> v >> u;
        v--; u--;
        g[v].push_back(u);
        g[u].push_back(v);
    }
    dfs(0);
    for (int av : ans)
        cout << av << ' ';
    return 0;
}
```

Task D (60)

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

using namespace std;

inline void split(string s) {
    //ifstream cin("1.txt");
    //ofstream cout("out.txt");
    string ch[3];
    ch[0] = s.substr(0, 3);
    ch[1] = s.substr(3, 3);
    ch[2] = s.substr(6, 3);
    string a[3];
    a[0] = "a" + ch[0] + ch[1] + " ";
    a[1] = "b" + ch[1] + ch[2] + " ";
    a[2] = "c" + ch[2] + ch[0];
    cout << a[0] + a[1] + a[2] << endl;
}

inline void merg(string a, string b) {
    //ifstream cin("1.txt");
    //ofstream cout("out.txt");
    string a1, a2, b1, b2;
    if (a > b)
        swap(a, b);
    a1 = a.substr(1, 3);
    a2 = a.substr(4, 3);
    b1 = b.substr(1, 3);
    b2 = b.substr(4, 3);
    //cout << b1 << "\n" << b2 << endl;
    if (a[0] == 'a' && b[0] == 'b') {
        cout << a1 + a2 + b2 << endl;
    }
    if (a[0] == 'a' && b[0] == 'c') {
        cout << a1 + a2 + b1 << endl;
    }
    if (a[0] == 'b' && b[0] == 'c') {
        cout << b2 + a1 + a2 << endl;
    }
}

inline void split2(string s) {
    //ifstream cin("1.txt");
    //ofstream cout("out.txt");
    string ch[3];
    ch[0] = s.substr(0, 3);
    ch[1] = s.substr(3, 3);
    ch[2] = s.substr(6, 3);
    string a[5];
    a[0] = "a" + ch[0] + ch[1] + " ";
    a[1] = "b" + ch[1] + ch[2] + " ";
    a[2] = "c" + ch[2] + ch[0] + " ";
    a[3] = a[0]; a[3][0] = 'd';
    a[4] = a[1]; a[4][0] = 'e';
    cout << a[0] + a[1] + a[2] + a[3] + a[4] << endl;
}

inline void merg2(string a, string b, string c) {
    //ifstream cin("1.txt");
    //ofstream cout("out.txt");
    string a1, a2, b1, b2;
    if (a[0] == 'd')
        a[0] = 'a';
    if (a[0] == 'e')
        a[0] = 'b';

    if (b[0] == 'd')
        b[0] = 'a';
    if (b[0] == 'e')
        b[0] = 'b';
```

```

    if (c[0] == 'd')
        c[0] = 'a';
    if (c[0] == 'e')
        c[0] = 'b';
    if (a[0] == b[0])
        swap(a, c);
    if (a > b)
        swap(a, b);
    a1 = a.substr(1, 3);
    a2 = a.substr(4, 3);
    b1 = b.substr(1, 3);
    b2 = b.substr(4, 3);
    //cout << b1 << "\n" << b2 << endl;
    if (a[0] == 'a' && b[0] == 'b') {
        cout << a1 + a2 + b2 << endl;
    }
    if (a[0] == 'a' && b[0] == 'c') {
        cout << a1 + a2 + b1 << endl;
    }
    if (a[0] == 'b' && b[0] == 'c') {
        cout << b2 + a1 + a2 << endl;
    }
}

int main()
{
    //ifstream cin("1.txt");
    //ofstream cout("out.txt");
    string tt;
    cin >> tt;
    int t, n, p;
    cin >> t >> n >> p;
    if (n == 3) {
        while (t--) {
            if (tt == "split") {
                string s;
                cin >> s;
                split(s);
            } else {
                string s1, s2;
                cin >> s1 >> s2;
                merg(s1, s2);
            }
        }
    } else {
        while (t--) {
            if (tt == "split") {
                string s;
                cin >> s;
                split2(s);
            } else {
                string s1, s2, s3;
                cin >> s1 >> s2 >> s3;
                merg2(s1, s2, s3);
            }
        }
    }
    return 0;
}

```

Task E (21)

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

using namespace std;

typedef long long ll;

struct vec{
    long double x, y;
    int ind;
    vec(){};
    vec(long double _x, long double _y) {
        x = _x;
        y = _y;
    }
    vec(const vec &a, const vec &b) {
        x = b.x - a.x;
        y = b.y - a.y;
    }
    vec ort() {
        return vec(y, -x);
    }
    long double len() {
        return sqrt(x * x + y * y);
    }
};

long double mv (const vec &a, const vec &b) {
    return a.x * b.y - a.y * b.x;
}

long double ms (const vec &a, const vec &b) {
    return a.x * b.x + a.y * b.y;
}

vec st;

bool cmp1(const vec &a, const vec &b) {
    if (a.x != b.x)
        return a.x < b.x;
    return a.y < b.y;
}

bool cmp2(const vec &a, const vec &b) {
    if (mv(vec(st, a), vec(st, b)) == 0) {
        return vec(st, a).len() < vec(st, b).len();
    }
    return mv(vec(st, a), vec(st, b)) > 0;
}

int main()
{
    //ifstream cin("1.txt");
    int n;
    cin >> n;
    vec P, Q;
    vector<vec> p(n);
    for (int i = 0; i < n; i++) {
        cin >> p[i].x >> p[i].y;
        // p[i].x *= 2; p[i].y *= 2;
        p[i].ind = i;
    }
    cin >> P.x >> P.y; //P.x *= 2; P.y *= 2;
    cin >> Q.x >> Q.y; //Q.x *= 2; Q.y *= 2;

    if (n == 2) {
        vec W = vec(p[0], p[1]).ort();
        vec R(P, Q);
        vec mid((p[0].x + p[1].x) / 2, (p[0].y + p[1].y) / 2);
        //cout << "!" << W.x << ' ' << W.y << endl;
    }
}
```

```

    //cout << "! " << R.x << ' ' << R.y << endl;
    if (mv(W, R) == 0) {
        if (mv(W, vec(P, mid)) == 0) {
            cout << -1;
        } else {
            if (mv(W, vec(P, mid)) > 0) {
                cout << 1;
            } else {
                if (mv(W, vec(P, mid)) < 0) {
                    cout << 2;
                }
            }
        }
    } else {
        if (mv(W, R) > 0) {
            cout << 2;
        } else {
            if (mv(W, R) < 0) {
                cout << 1;
            }
        }
    }
    return 0;
}

vec R(P, Q);
sort(p.begin(), p.end(), cmp1);
st = p[0];
sort(p.begin() + 1, p.end(), cmp2);
vector<vec> wr;
wr.push_back(p[0]);
wr.push_back(p[1]);
for (int i = 2; i < n; i++) {
    int sz = wr.size();
    while (sz > 1 && mv(
        vec(wr[sz - 2], wr[sz - 1]),
        vec(wr[sz - 1], p[i])) < 0) {
        wr.pop_back();
        sz--;
    }
    wr.push_back(p[i]);
}
long double kek = 0;
long double kek2 = 0;
for (int i = 0; i < wr.size(); i++) {
    vec a = wr[i];
    vec b = wr[(i + 1) % wr.size()];
    vec c = wr[(i + 2) % wr.size()];
    vec W1 = vec(a, b).ort();
    kek += abs(mv(vec(a, b), vec(b, c)));
    if (mv(W1, R) != 0)
        kek2 = 1;
}

if (kek == 0 && kek2 == 0) {
    long double minn = 1e19;
    long double lol = -1;
    for (int i = 0; i < wr.size(); i++) {
        vec a = wr[i];
        vec b = wr[(i + 1) % wr.size()];
        vec W1 = vec(a, b).ort();
        vec mid = vec((a.x + b.x) / 2, (a.y + b.y) / 2);
        if (mv(W1, vec(mid, P)) == 0) {
            cout << -1;
            return 0;
        }
    }
}
for (int i = 0; i < wr.size(); i++) {
    if (minn > vec(wr[i], P).len()) {
        minn = vec(wr[i], P).len();
        lol = wr[i].ind + 1;
    }
}
cout << lol;
return 0;
}

```

```

for (int i = 0; i < wr.size(); i++) {
    vec a = wr[i];
    vec b = wr[(i + 1) % wr.size()];
    vec c = wr[(i + 2) % wr.size()];
    vec W1 = vec(a, b).ort();
    vec W2 = vec(b, c).ort();
    if (mv(W1, R) > 0 && mv(R, W2) > 0) {
        cout << b.ind + 1;
        return 0;
    }
}
//cout << "loool";
P.x += R.x * 1e20;
P.y += R.y * 1e20;
for (int i = 0; i < wr.size(); i++) {
    vec a = wr[i];
    vec b = wr[(i + 1) % wr.size()];
    vec W = vec(a, b).ort();
    kek += ms(a, b);
    //cout << a.ind + 1 << ' ' << b.ind + 1 << ' ' << mv(R, W) << ' ' << ms(R, W) << endl;
    vec mid = vec((a.x + b.x) / 2, (a.y + b.y) / 2);
    if (mv(R, W) == 0 && (!kek || ms(R, W) > 0)) {
        if (mv(W, vec(mid, P)) == 0) {
            cout << -1;
            return 0;
        }
        if (mv(W, vec(mid, P)) < 0) {
            cout << a.ind + 1;
            return 0;
        }
        if (mv(W, vec(mid, P)) > 0) {
            cout << b.ind + 1;
            return 0;
        }
    }
}
return 0;
}

```


Task F (—)