

Олимпиада СПбГУ по информатике 2018/19 учебного года

Макогон Артём Аркадьевич

A	B	C	D	E	F	Sum
100	100	100	60	100	7	467

Task A (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld long double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5;

ll d[maxn];

int f(int i) {
    return i & (i + 1);
}

void add(int ind, int val) {
    while (ind < maxn) {
        d[ind] += val;
        ind |= (ind + 1);
    }
}

ll prefsum(int i) {
    ll ans = 0;
    while (i >= 0) {
        ans += d[i];
        i = f(i) - 1;
    }
    return ans;
}

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
```

```
cin.tie(0); cout.tie(0);

int n, m;
cin >> n >> m;
while (n < m)
    n *= 2;
if (n == m) {
    cout << "Yes";
}
else
    cout << "No";
return 0;
}
```

Task B (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld long double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5;

ll d[maxn];

int f(int i) {
    return i & (i + 1);
}

void add(int ind, int val) {
    while (ind < maxn) {
        d[ind] += val;
        ind |= (ind + 1);
    }
}

ll prefsum(int i) {
    ll ans = 0;
    while (i >= 0) {
        ans += d[i];
        i = f(i) - 1;
    }
    return ans;
}

string s;

bool check(int i, int j) {
    for (int k = i; k < j; k++) {
        if (s[k] == 'o' && s[k + 1] == 'r')
            return true;
    }
    return false;
}

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    int n;
    cin >> n;

    cin >> s;
    if (check(0, s.size() - 1)) {
        cout << "Yes";
        return 0;
    }
}
```

```

    }

    for (int i = 0; i < s.size() - 1; i++) {
        swap(s[i], s[i + 1]);
        if (check(max(i - 1, 0), min(int(s.size()) - 1, i + 3))) {
            cout << "Yes";
            return 0;
        }
        swap(s[i], s[i + 1]);
    }
    cout << "No";
    return 0;
}

```

Task C (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld long double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5 + 10;

ll d[maxn];

int f(int i) {
    return i & (i + 1);
}

void add(int ind, int val) {
    while (ind < maxn) {
        d[ind] += val;
        ind |= (ind + 1);
    }
}

ll prefsum(int i) {
    ll ans = 0;
    while (i >= 0) {
        ans += d[i];
        i = f(i) - 1;
    }
    return ans;
}

string s;

bool check(int i, int j) {
    for (int k = i; k < j; k++) {
        if (s[k] == 'o' && s[k + 1] == 'r')
            return true;
    }
    return false;
}

vector<vector<int>>> g;
vector<vector<int>>> sz;
vector<int> sumsz;
bool used[maxn];

void dfs(int v, int p = -1) {
    used[v] = true;
    for (auto ev : g[v]) {
        if (used[ev])
            continue;
    }
}
```

```

        dfs(ev);
        sumsz[v] += sumsz[ev];
        sz[v].pb(sumsz[ev]);
    }
    sumsz[v]++;
}

int main() {
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    int n;
    cin >> n;
    g.resize(n + 1);
    sz.resize(n + 1);
    sumsz.resize(n + 1);

    for (int i = 0; i < n - 1; i++) {
        int u, v;
        cin >> u >> v;
        g[u].pb(v);
        g[v].pb(u);
    }

    dfs(1);

    for (int i = 1; i <= n; i++) {
        sz[i].pb(n - sumsz[i]);
    }
    for (int i = 1; i <= n; i++)
        sort(all(sz[i]));
    for (int i = 1; i <= n; i++)
        cout << sz[i].back() + 1 << " ";

    return 0;
}

```

Task D (60)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld long double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5 + 10;

ll d[maxn];

int f(int i) {
    return i & (i + 1);
}

void add(int ind, int val) {
    while (ind < maxn) {
        d[ind] += val;
        ind |= (ind + 1);
    }
}

ll prefsum(int i) {
    ll ans = 0;
    while (i >= 0) {
        ans += d[i];
        i = f(i) - 1;
    }
    return ans;
}

unordered_multiset<char> bar;

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    srand(time(NULL));

    string sr;
    cin >> sr;
    int n, t, p;
    cin >> t >> n >> p;

    for (int w = 0; w < t; w++) {
        if (sr == "split") {
            if (n == 3) {
                string s;
                cin >> s;
                string s1, s2, s3;
                s1 += 'l';
            }
        }
    }
}
```

```

        for (int i = 0; i < 6; i++)
            s1 += s[i];
        s2 += 'r';
        for (int i = 3; i < 9; i++)
            s2 += s[i];
        s3.resize(7);
        s3[0] = 'm';
        s3[1] = s[0];
        s3[2] = s[1];
        s3[3] = s[2];
        s3[4] = s[6];
        s3[5] = s[7];
        s3[6] = s[8];

        cout << s1 << " " << s2 << " " << s3 << "\n";
    }
    if (n == 5) {
        string s;
        cin >> s;
        string s1, s2, s3;
        s1 += 'l';
        for (int i = 0; i < 6; i++)
            s1 += s[i];
        s2 += 'r';
        for (int i = 3; i < 9; i++)
            s2 += s[i];
        s3.resize(7);
        s3[0] = 'm';
        s3[1] = s[0];
        s3[2] = s[1];
        s3[3] = s[2];
        s3[4] = s[6];
        s3[5] = s[7];
        s3[6] = s[8];

        cout << s1 << " " << s1 << " " << s2 << " " << s2 << " " << s3 <<
            "\n";
    }
}
else {
    if (n == 3) {
        string s1, s2, res;
        cin >> s1 >> s2;
        if (s2[0] == 'l')
            swap(s1, s2);
        if (s1[0] == 'r')
            swap(s2, s1);
        if (s1[0] == 'l')
            for (int i = 1; i < s1.size(); i++)
                res += s1[i];
        else
            for (int i = 1; i < 4; i++)
                res += s1[i];

        for (int i = s2.size() - 9 + res.size(); i < s2.size(); i++)
            res += s2[i];
        cout << res << "\n";
    }
    if (n == 5) {
        string s1, s2, s3, res;
        cin >> s1 >> s2 >> s3;
        if (s1[0] == s2[0])
            swap(s2, s3);

        if (s2[0] == 'l')
            swap(s1, s2);
        if (s1[0] == 'r')
            swap(s2, s1);
        if (s1[0] == 'l')
            for (int i = 1; i < s1.size(); i++)
                res += s1[i];
        else
            for (int i = 1; i < 4; i++)
                res += s1[i];
    }
}

```



```

        for (int i = s2.size() - 9 + res.size(); i < s2.size(); i++)
            res += s2[i];
        cout << res << "\n";
    }

}
return 0;
}

```

Task E (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld long double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5 + 10;
const ld eps = 1e-8;
const ld inf = 1e17;
const ld gmt = 7;

struct Point{
    ld x, y;
    Point() {}
    Point(ld a, ld b) {
        x = a;
        y = b;
    }
};

struct Line {
    ld a, b, c;
    Line() {}
    Line(ld x, ld y, ld z) {
        a = x;
        b = y;
        c = z;
    }

    Line(Point p1, Point p2) {
        a = p2.y - p1.y;
        b = p1.x - p2.x;
        c = -1 * (a * p1.x + b * p1.y);
    }
};

ld sqr(ld x) {
    return x * x;
}

ld distpl(Point p, Line l) {
    ld ans = 0;
    ans = (l.a * p.x + l.b * p.y + l.c) / (sqrt(sqr(l.a) + sqr(l.b)));
    return ans;
}

ld distp(Point p1, Point p2) {
    return sqrt(sqr(p1.x - p2.x) + sqr(p1.y - p2.y));
}

ld len(Point p) {
```

```

        return sqrt(sqr(p.x) + sqr(p.y));
    }

Point per(Line l1, Line l2) {
    ld x = (l1.b * l2.c - l2.b * l1.c) / (l1.a * l2.b - l2.a * l1.b);
    ld y = (l1.a * l2.c - l2.a * l1.c) / (l2.a * l1.b - l1.a * l2.b);
    return Point(x, y);
}

Point ph(Point p, Line l) {
    Line vsp = Line(p, Point(p.x + l.a, p.y + l.b));

    Point res = per(l, vsp);

    //ld d = distpl(p, l);
    //Point h = Point(l.a, l.b);
    //ld curlen = len(h);
    //h.x *= (d / curlen);
    //h.y *= (d / curlen);

    //Point res = Point(p.x - h.x, p.y - h.y);

    //if (fabs(l.a * res.x + l.b * res.y + l.c) > eps)
    //    exit(-1);
    return res;
}

vector<Point> foo;
vector<Point> bar;

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        ld x, y;
        cin >> x >> y;
        x *= gmt;
        y *= gmt;
        foo.emplace_back(x, y);
    }

    ld x, y;
    cin >> x >> y;
    x *= gmt;
    y *= gmt;
    Point p = Point(x, y);
    cin >> x >> y;
    x *= gmt;
    y *= gmt;
    Point q = Point(x, y);

    Line l = Line(p, q);

    for (auto ev : foo) {
        bar.pb(ph(ev, l));
    }

    ld maxd = -inf;
    int ans = 0;
    bool flag = false;

    for (int i = 0; i < bar.size(); i++) {
        Point cur = bar[i];
        ld mn = 1;

```

```

Line vsp = Line(p, Point(p.x + l.a, p.y + l.b));

if (distpl(cur, vsp) * distpl(q, vsp) < 0)
    mn = -1;

ld curdis = distp(p, cur) * mn;

if (fabs(curdis - maxd) < eps) {
    if (fabs(fabs(distpl(foo[i], l)) - fabs(distpl(foo[ans], l))) < eps)
        flag = true;
    else {
        if (fabs(distpl(foo[i], l)) < fabs(distpl(foo[ans], l))) {
            ans = i;
            flag = false;
        }
    }
}
if (curdis > maxd) {
    maxd = curdis;
    ans = i;
    flag = false;
}
}

if (flag)
    cout << -1;
else
    cout << ans + 1;

return 0;
}

```

Task F (7)

```
#define _CRT_SECURE_NO_WARNINGS
#include<iostream>
#include<vector>
#include<set>
#include<algorithm>
#include<map>
#include<string>
#include<unordered_set>
#include<iomanip>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ld double
#define ff first
#define ss second
#define pb push_back

using namespace std;

const int maxn = 1e5 + 10;
const ld eps = 1e-10;
const ld T0 = 10000;

ld T = T0;
ld a = 0.999;

vector<int> r, b;

int n, k;

ll h(ll mask) {
    int curr = 0;
    int curb = 0;
    int curans = 0;
    for (int i = 0; i < n; i++) {
        if (mask == (mask | (1 << i)))
            curr += k;
        else
            curb += k;
        curans += min(curb, b[i]);
        curb = max(0, curb - b[i]);

        curans += min(curr, r[i]);
        curr = max(0, curr - r[i]);
    }

    return curans;
}

int main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    srand(time(NULL));

    cin >> n >> k;

    for (int i = 0; i < n; i++) {
        int q, w;
        cin >> q >> w;
        r.pb(q);
        b.pb(w);
    }
}
```

```

ll mask = rand() % (1 << n);

ll res = 0;

if (n < 20) {
    for (int mask = 0; mask <= (1 << n) - 1; mask++) {
        int curr = 0;
        int curb = 0;
        int curans = 0;
        for (int i = 0; i < n; i++) {
            if (mask == (mask | (1 << i)))
                curr += k;
            else
                curb += k;
            curans += min(curb, b[i]);
            curb = max(0, curb - b[i]);

            curans += min(curr, r[i]);
            curr = max(0, curr - r[i]);
        }
        res = max(int(res), curans);
    }
    cout << res;
    return 0;
}

for (int i = 0; i < 100; i++) {
    while (T > eps) {
        ll h1 = h(mask);
        int q = rand() % n;
        mask ^= (1 << q);
        ll h2 = h(mask);

        if (h2 < h1) {
            ld ran = ld(rand());
            while (ran > 1)
                ran /= 10;

            if (exp(-(h2 - h1) / T) > ran)
                mask ^= (1 << q);
        }

        T *= a;
    }
    res = max(res, h(mask));
    T = T0;
}
cout << res;
return 0;
}

```