

Олимпиада СПбГУ по информатике 2018/19 учебного года

Титов Тимофей Андреевич

A	B	C	D	E	F	Sum
100	100	100	60	100	0	460

Task A (100)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>

using namespace std;
using ll = long long;
using ld = long double;

int main() {
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);

    int a, b;
    cin >> a >> b;
    while (b > a){
        if (b % 2 == 1){
            cout << "NO";
            return 0;
        }
        b /= 2;
    }
    if (a == b){
        cout << "YES";
    } else {
        cout << "NO";
    }

    return 0;
}
```

Task B (100)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>
#include <string>

using namespace std;
using ll = long long;
using ld = long double;

int main() {
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);

    int n;
    cin >> n;
    string s;
    cin >> s;

    for (int i = 0; i < n-2; i++){
        if (s[i] == 'o'){
            if ((s[i+1] == 'r') || (s[i+2] == 'r')){
                cout << "Yes";
                return 0;
            }
            if ((i > 0) && (s[i-1] == 'r')){
                cout << "Yes";
                return 0;
            }
        }
    }

    if ((s[n-2] == 'o') && ((s[n-1] == 'r') || ((n > 2) && (s[n-3] == 'r')))){
        cout << "Yes";
        return 0;
    }
    if ((s[n-1] == 'o') && (s[n-2] == 'r')){
        cout << "Yes";
        return 0;
    }

    cout << "No";

    return 0;
}
```

Task C (100)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>

using namespace std;
using ll = long long;
using ld = long double;

int n;
vector<vector<int>>> rebr;
vector<bool> used;
vector<int> kol;
vector<int> ans;

int fillkol(int a){
    used[a] = true;
    int res = 1;
    for (int i : rebr[a]){
        if (!used[i]){
            res += fillkol(i);
        }
    }
    kol[a] = res;
    return res;
}

void fillans(int a){
    used[a] = true;
    for (int i : rebr[a]){
        if (used[i])
            continue;
        int maxim = n - kol[i];
        for (int j : rebr[i]){
            if (used[j])
                continue;
            if (kol[j] > maxim){
                maxim = kol[j];
            }
        }
        ans[i] = maxim + 1;
        // cout << i << " " << ans[i] << "\n";
        fillans(i);
    }
}

int main() {
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);

    cin >> n;
    rebr = vector<vector<int>>>(n);
    used = vector<bool>(n, false);
    kol = vector<int>(n);
    ans = vector<int>(n);

    for (int i = 0; i < n-1; i++){
        int a, b;
        cin >> a >> b;
        a--; b--;
        rebr[a].push_back(b);
        rebr[b].push_back(a);
    }

    fillkol(0);
```

```

int maxim = 0;
for (int i : rebr[0]){
    if (kol[i] > maxim){
        maxim = kol[i];
    }
}
ans[0] = maxim+1;

used = vector<bool>(n, false);
fillans(0);

for (int i = 0; i < n; i++){
    cout << ans[i] << " ";
}

return 0;
}

```

Task D (60)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>

using namespace std;
using ll = long long;
using ld = long double;

int t, n, p;
string d;

void split3(string s){
    cout << "a" + s.substr(0,6) << "_" << "b" + s.substr(3, 6) << "_" << "c" + s.substr(0,3) + s.
        substr(6,3);
}

void merge2(string s1, string s2){
    if (s1[0] == 'a'){
        cout << s1.substr(1, 6) + s2.substr(4,3);
        return;
    }
    if (s2[0] == 'a'){
        cout << s2.substr(1, 6) + s1.substr(4,3);
        return;
    }
    if (s1[0] == 'b'){
        cout << s2.substr(1,3) + s1.substr(1,6);
    } else {
        cout << s1.substr(1,3) + s2.substr(1,6);
    }
}

void split5(string s){
    cout << "a" + s.substr(0,6) << "_" << "a" + s.substr(0,6) << "_" << "b" + s.substr(3, 6) << "_"
        << "b" + s.substr(3, 6) << "_" << "c" + s.substr(0,3) + s.substr(6,3);
}

void merge3(string s1, string s2, string s3){
    if (s1[0] == 'a'){
        if (s2[0] != 'a') {
            cout << s1.substr(1, 6) + s2.substr(4, 3);
        } else {
            cout << s1.substr(1, 6) + s3.substr(4, 3);
        }
        return;
    }
    if (s2[0] == 'a'){
        if (s1[0] != 'a') {
            cout << s2.substr(1, 6) + s1.substr(4, 3);
        } else {
            cout << s2.substr(1, 6) + s3.substr(4, 3);
        }
        return;
    }
    if (s1[0] == 'b'){
        if (s2[0] != 'b') {
            cout << s2.substr(1, 3) + s1.substr(1, 6);
        } else {
            cout << s3.substr(1, 3) + s1.substr(1, 6);
        }
    } else {
        cout << s1.substr(1,3) + s2.substr(1,6);
    }
}

int main() {
```

```

ios::sync_with_stdio(0);
cout.tie(0);
cin.tie(0);

cin >> d >> t >> n >> p;
if ((d == "split") && (n == 3)) {
    for (int i = 0; i < t; i++) {
        string s;
        cin >> s;
        split3(s);
        cout << "\n";
    }
}
if ((d == "merge") && (n == 3)) {
    for (int i = 0; i < t; i++) {
        string s1, s2;
        cin >> s1 >> s2;
        merge2(s1, s2);
        cout << "\n";
    }
}

if ((d == "split") && (n == 5)) {
    for (int i = 0; i < t; i++) {
        string s;
        cin >> s;
        split5(s);
        cout << "\n";
    }
}
if ((d == "merge") && (n == 5)) {
    for (int i = 0; i < t; i++) {
        string s1, s2, s3;
        cin >> s1 >> s2 >> s3;
        merge3(s1, s2, s3);
        cout << "\n";
    }
}

return 0;
}

/*
apasswo bswords cpasrds
auhaaaa baaaaaa cuhaaaa
aaaaaaa baaaaaa caaaaaa
aplainw binword cplaord

*/

```

Task E (100)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>

using namespace std;
using ll = long long;
using ld = long double;

int n;
int xp, yp, xq, yq;
vector<pair<int, int>> gos;
ld k;
ld ex = 1e-9;

ld xk(int i){
    return gos[i].second + gos[i].first/k;
}

ld yk(int i){
    return gos[i].second - gos[i].first*k;
}

int main() {
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);

    cin >> n;
    gos = vector<pair<int, int>>(n);

    for (int i = 0; i < n; i++){
        cin >> gos[i].first >> gos[i].second;
    }

    cin >> xp >> yp >> xq >> yq;

    if (xp == xq){
        if (yp < yq){
            int maxim = -1e9 - 10;
            int maxi;
            bool maxeq = false;
            for (int i = 0; i < n; i++){
                if (gos[i].second == maxim){
                    if (abs(gos[i].first - xp) == abs(gos[maxi].first - xp)){
                        maxeq = true;
                    }
                    if (abs(gos[i].first - xp) < abs(gos[maxi].first - xp)){
                        maxi = i;
                        maxeq = false;
                    }
                }
                if (gos[i].second > maxim){
                    maxim = gos[i].second;
                    maxi = i;
                    maxeq = false;
                }
            }
            if (maxeq){
                cout << -1;
                return 0;
            } else {
                cout << maxi + 1;
                return 0;
            }
        } else {
            int maxim = 1e9 + 10;
```

```

int maxi;
bool maxeq = false;
for (int i = 0; i < n; i++){
    if (gos[i].second == maxim){
        if (abs(gos[i].first - xp) == abs(gos[maxi].first - xp)){
            maxeq = true;
        }
        if (abs(gos[i].first - xp) < abs(gos[maxi].first - xp)){
            maxi = i;
            maxeq = false;
        }
    }
    if (gos[i].second < maxim){
        maxim = gos[i].second;
        maxi = i;
        maxeq = false;
    }
}
if (maxeq){
    cout << -1;
    return 0;
} else {
    cout << maxi + 1;
    return 0;
}
}

k = ((ld)yq - yp)/(xq - xp);

if (k == 0){
    if (xp < xq){
        int maxim = -1e9 - 10;
        int maxi;
        bool maxeq = false;
        for (int i = 0; i < n; i++){
            if (gos[i].first == maxim){
                if (abs(gos[i].second - yp) == abs(gos[maxi].second - yp)){
                    maxeq = true;
                }
                if (abs(gos[i].second - yp) < abs(gos[maxi].second - yp)){
                    maxi = i;
                    maxeq = false;
                }
            }
            if (gos[i].first > maxim){
                maxim = gos[i].first;
                maxi = i;
                maxeq = false;
            }
        }
        if (maxeq){
            cout << -1;
            return 0;
        } else {
            cout << maxi + 1;
            return 0;
        }
    } else {
        int maxim = 1e9 + 10;
        int maxi;
        bool maxeq = false;
        for (int i = 0; i < n; i++){
            if (gos[i].first == maxim){
                if (abs(gos[i].second - yp) == abs(gos[maxi].second - yp)){
                    maxeq = true;
                }
                if (abs(gos[i].second - yp) < abs(gos[maxi].second - yp)){
                    maxi = i;
                    maxeq = false;
                }
            }
        }
        if (gos[i].first < maxim){
            maxim = gos[i].first;

```



```

        maxi = i;
        maxeq = false;
    }
}
if (maxeq){
    cout << -1;
    return 0;
} else {
    cout << maxi + 1;
    return 0;
}
}
}

ld y0 = yq - xq*k;

if (yq + xq/k > yp + xp/k){
    ld maxim = -1e18 - 10;
    int maxi;
    bool maxeq = false;
    for (int i = 0; i < n; i++){
        if (abs(xk(i) - maxim) < ex){
            if (abs(abs(yk(i) - y0) - abs(yk(maxi) - y0)) < ex){
                maxeq = true;
            } else
            if (abs(yk(i) - y0) < abs(yk(maxi) - y0)){
                maxi = i;
                maxeq = false;
            }
        } else
        if (xk(i) > maxim){
            maxim = xk(i);
            maxi = i;
            maxeq = false;
        }
    }
    if (maxeq){
        cout << -1;
        return 0;
    } else {
        cout << maxi + 1;
        return 0;
    }
} else {
    ld maxim = 1e18 + 10;
    int maxi;
    bool maxeq = false;
    for (int i = 0; i < n; i++){
        if (abs(xk(i) - maxim) < ex){
            if (abs(abs(yk(i) - y0) - abs(yk(maxi) - y0)) < ex){
                maxeq = true;
            } else
            if (abs(yk(i) - y0) < abs(yk(maxi) - y0)){
                maxi = i;
                maxeq = false;
            }
        } else
        if (xk(i) < maxim){
            maxim = xk(i);
            maxi = i;
            maxeq = false;
        }
    }
    if (maxeq){
        cout << -1;
        return 0;
    } else {
        cout << maxi + 1;
        return 0;
    }
}

return 0;

```


Task F (0)

```
#include <iostream>
#include <algorithm>
#include <cmath>
#include <vector>
#include <map>
#include <queue>

using namespace std;
using ll = long long;
using ld = long double;

int n, k;
vector<int> r, b;
ll ans = 0, sumr = 0, sumb = 0;
int kolr = 0, kolb = 0;

int main() {
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);

    cin >> n >> k;
    r = vector<int>(n);
    b = vector<int>(n);

    for (int i = 0; i < n; i++){
        cin >> r[i] >> b[i];
        ans += r[i] + b[i];
        sumr += r[i];
        sumb += b[i];
    }

    vector<int> nol = {0};

    for (int i = 0; i < n; i++){
        if ((r[i] - kolr > b[i] - kolb)&&((sumr - kolr >= k)||((sumr - kolr > sumb - kolb)|| (kolr
            + k - sumr < b[i] - kolb - k)))){
            // if ((r[i] - kolr > b[i] - kolb)&&((sumr - kolr >= k)
            // ||((max((ll)0, kolr + k - sumb) > max((ll)0, kolb + k - sumb))))){
            ans -= max(0, r[i] - kolr - k) + max(0, b[i] - kolb);
            kolb = max(0, kolb - b[i]);
            kolr = max(0, kolr + k - r[i]);
        } else {
            ans -= max(0, b[i] - kolb - k) + max(0, r[i] - kolr);
            kolb = max(0, kolb + k - b[i]);
            kolr = max(0, kolr - r[i]);
        }
        sumb -= b[i];
        sumr -= r[i];
    }

    cout << ans;

    return 0;
}
```