

Олимпиада СПбГУ по информатике 2018/19 учебного года

Сурков Петр Анатольевич

A	B	C	D	E	F	Sum
100	100	100	60	100	24	484

Task A (100)

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

using namespace std;

bool isPow2(int x)
{
    if (x == 1)
        return true;
    int a = 1;
    while (a < x)
    {
        a *= 2;
        if (a == x)
            return true;
    }
    return false;
}

int main()
{
    int a, b;
    cin >> a >> b;
    if (b % a != 0 || isPow2(b / a) == 0)
        cout << "No";
    else
        cout << "Yes";

    return 0;
}
```

Task B (100)

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

using namespace std;

int main()
{
    int n;
    cin >> n;
    string s;
    cin >> s;
    bool f = false;
    for (int i = 0; i < n - 1; i++)
    {
        if (s[i] == 'o' && s[i + 1] == 'r')
            f = true;
        if (s[i] == 'r' && s[i + 1] == 'o')
            f = true;
    }

    for (int i = 0; i < n - 2; i++)
        if (s[i] == 'o' && s[i + 2] == 'r')
            f = true;
    if (f)
        cout << "Yes";
    else
        cout << "No";

    return 0;
}
```

Task C (100)

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

using namespace std;

vector<vector<int>>> g;
vector<int> was;
vector<int> sz;
vector<int> maxsz;

void dfs1(int v)
{
    was[v] = 1;
    sz[v] = 1;
    for (auto u : g[v])
    {
        if (was[u] == 0)
        {
            dfs1(u);
            sz[v] += sz[u];
        }
    }
}

void dfs2(int v)
{
    was[v] = 1;
    int sum_sz = 0;
    for (auto u : g[v])
    {
        if (was[u] == 0)
        {
            dfs2(u);
            sum_sz += sz[u];
            maxsz[v] = max(maxsz[v], sz[u]);
        }
    }
    maxsz[v] = max(maxsz[v], sz[v] - sum_sz - 1);
}

int main()
{
    int n;
    cin >> n;
    g.resize(n);
    was.resize(n);
    sz.resize(n);
    maxsz.resize(n);
    int v, u;
    for (int i = 0; i < n - 1; i++)
    {
        cin >> v >> u;
        v--;
        u--;
        g[v].push_back(u);
        g[u].push_back(v);
    }
    dfs1(0);
    fill(was.begin(), was.end(), 0);
    dfs2(0);
    for (auto i : maxsz)
        cout << i + 1 << ' ';

    return 0;
}
```

Task D (60)

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

using namespace std;

int main()
{
    string s;
    cin >> s;
    int t, n, p;
    cin >> t >> n >> p;
    if (n == 3)
    {
        if (s[0] == 's')
        {
            for (int i = 0; i < t; i++)
            {
                cin >> s;
                cout << 'a';
                for (int j = 0; j < 6; j++)
                    cout << s[j];
                cout << '␣';

                cout << 'b';
                for (int j = 0; j < 6; j++)
                    cout << s[3 + j];
                cout << '␣';

                cout << 'c';
                for (int j = 0; j < 6; j++)
                    cout << s[(6 + j) % 9];
                cout << endl;
            }
        }
        else
        {
            for (int i = 0; i < t; i++)
            {
                string res = "000000000";
                for (int j = 0; j < (n + 1) / 2; j++)
                {
                    cin >> s;
                    int st = (s[0] - 'a') * 3;
                    for (int i = 0; i < 6; i++)
                        res[(i + st) % 9] = s[i + 1];
                }
                cout << res << endl;
            }
        }
    }
    else if (n == 5)
    {
        if (s[0] == 's')
        {
            for (int i = 0; i < t; i++)
            {
                cin >> s;
                for (int z = 0; z < 5; z++)
                {
                    cout << char(z + 'a');
                    for (int j = 0; j < 6; j++)
                        cout << s[(2 * z + j) % 9];
                    if (z != 4)
                        cout << '␣';
                    else
                        cout << endl;
                }
            }
        }
    }
}
```

```

        }
    }
}
else
{
    for (int i = 0; i < t; i++)
    {
        string res = "000000000";
        for (int j = 0; j < (n + 1) / 2; j++)
        {
            cin >> s;
            int st = (s[0] - 'a') * 2;
            for (int i = 0; i < 6; i++)
                res[(i + st) % 9] = s[i + 1];
        }
        cout << res << endl;
    }
}

}

return 0;
}

```

Task E (100)

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

using namespace std;

struct Vec
{
    long long x, y;
};

Vec operator*(const Vec &a, const long long k)
{
    return Vec({a.x * k, a.y * k});
}

Vec operator+(const Vec &a, const Vec &b)
{
    return Vec({a.x + b.x, a.y + b.y});
}

Vec operator-(const Vec &a, const Vec &b)
{
    return Vec({a.x - b.x, a.y - b.y});
}

long long operator*(const Vec &a, const Vec &b)
{
    return a.x * b.x + a.y * b.y;
}

long long len2(Vec a)
{
    return a * a;
}

Vec P, Q;

bool cmp(pair<Vec, int> &a, pair<Vec, int> &b)
{
    long long v1 = (Q - P) * (a.first - P), v2 = (Q - P) * (b.first - P);
    //cout << v1 << ' ' << v2 << endl;
    if (v2 != v1)
        return v2 - v1 < 0;
    return len2(a.first - P) < len2(b.first - P);
}

int main()
{
    int n;
    cin >> n;
    if (n == 1)
    {
        cout << 1;
        return 0;
    }
    vector<pair<Vec, int>> g(n);
    for (int i = 0; i < n; i++)
    {
        cin >> g[i].first.x >> g[i].first.y;
        g[i].second = i;
    }
    cin >> P.x >> P.y >> Q.x >> Q.y;

    sort(g.begin(), g.end(), cmp);
    if (cmp(g[0], g[1]))
        cout << g[0].second + 1;
```

```
    else
        cout << -1;
    return 0;
}
```

Task F (24)

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

using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n, k, all_a = 0, all_b = 0;
    cin >> n >> k;
    vector<pair<int, int>> v(n);
    for (auto &i : v)
        cin >> i.first >> i.second;
    for (int i = n - 1; i >= 0; i--)
    {
        int need = min(n * k - all_a, v[i].first);
        v[i].first = need;
        all_a += need;

        need = min(n * k - all_b, v[i].second);
        v[i].second = need;
        all_b += need;
    }

    vector<int> dayA(all_a), dayB(all_b);

    int cur_a = 0, cur_b = 0;
    for (int i = n - 1; i >= 0; i--)
    {
        for (int j = 0; j < v[i].first; j++)
            dayA[cur_a + j] = i;
        cur_a += v[i].first;

        for (int j = 0; j < v[i].second; j++)
            dayB[cur_b + j] = i;
        cur_b += v[i].second;
    }

    int bin_l = 0, bin_r = all_a + all_b + 1, bin_m;
    vector<pair<int, int>> bake_in(n);
    while (bin_l + 1 < bin_r)
    {
        bin_m = (bin_l + bin_r) / 2;
        bool can = false;
        int st = 0;
        if (bin_m - all_a > 0)
            st = bin_m - all_a;
        for (int b = all_b; b >= st; b--)
        {
            int a = bin_m - b;

            // fill(bake_in.begin(), bake_in.end(), 0);
            for (auto &p : bake_in)
                p.first = p.second = 0;

            int t = a - 1;
            while (t >= 0) // O(n)
            {
                bake_in[dayA[t]].first++;
                t -= k;
            }
            t = b - 1;
            while (t >= 0) // O(n)
            {

```



```

        bake_in[dayB[t]].second++;
        t -= k;
    }

    int nbake_a = 0, nbake_b = 0;
    bool ok = true;
    for (int i = 0; i < n; i++) // check, O(n)
    {
        nbake_a += bake_in[i].first;
        nbake_b += bake_in[i].second;
        if (nbake_a + nbake_b > i + 1)
        {
            ok = false;
            break;
        }
    }
    if (ok)
    {
        can = true;
        break;
    }
}
if (can)
    bin_l = bin_m;
else
    bin_r = bin_m;
}
cout << bin_l;

return 0;
}

```