Олимпиада СПбГУ по информатике 2018/19 учебного года

Макаров Владимир Владимирович

A	B	C	D	E	F	Sum
100	100	100	0	100	0	400

Task A (100)

```cpp
#include<iostream>
#include<cmath>

using namespace std;

long long n, m, d, x;

int main()
{
    cin >> n >> m;
    d = m / n;
    for(x = 1; x < d; x *= 2);
    //cout << d << ' ' << x << endl;
    if(d > 0 && m % n == 0 && x == d)
        cout << "Yes";
    else
        cout << "No";
    return 0;
}
```

Task B (100)

```cpp
#include<iostream>
#include<string>

using namespace std;

string s;
int n;

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

Task C (100)

```cpp
#include<iostream>
#include<vector>

using namespace std;

const int N = 100005;

vector<int> g[N];
int n, x, y, kv[N], is[N], kmax;

int dfs(int v)
{
    is[v] = 1;
    if(v != 1 && g[v].size() == 1)
    {
        kv[v] = 1;
        //cout << v << endl;
        return 1;
    }
    if(n == 1)
    {
        kv[v] = 1;
        return 1;
    }

    kv[v] = 1;
    for(int i = 0; i < g[v].size(); ++i)
    {
        if(!is[g[v][i]])
            kv[v] += dfs(g[v][i]);
    }
    return kv[v];
}

int main()
{
    //freopen("input.txt", "r", stdin);
    cin >> n;
    for(int i = 1; i < n; ++i)
    {
        cin >> x >> y;
        g[x].push_back(y);
        g[y].push_back(x);
    }
    dfs(1);
    /*cout << endl << endl;
    for(int i = 1; i <= n; ++i)
        cout << i << " - " << kv[i] << endl;*/
    for(int i = 1; i <= n; ++i)
    {
        kmax = kv[1] - kv[i];
        for(int j = 0; j < g[i].size(); ++j)
            if(kv[g[i][j]] > kmax && kv[i] > kv[g[i][j]])
                kmax = kv[g[i][j]];
        cout << kmax + 1 << endl;
    }
    return 0;
}
```

Task D (—)

Task E (100)

```cpp
#include<iostream>
#include<vector>

using namespace std;

int n;
int x[100005], y[100005];
long long xp, yp, xq, yq, xv, yv, imax;
long long A, B, C, SAB, a, b, c, d, dmax, dq;

int main()
{
    //freopen("input.txt", "r", stdin);
    cin >> n;
    for(int i = 0; i < n; ++i)
    {
        cin >> x[i] >> y[i];
    }
    cin >> xp >> yp >> xq >> yq;
    A = yq - yp;
    B = xp - xq;
    C = yp * (xq - xp) - xp * (yq - yp);

    //cout << A << ' ' << B << ' ' << C << endl;
    //SAB = A * A + B * B;
    b = (-1) * A;
    a = B;
    c = (-1) * B * xp + A * yp;

    //cout << a << ' ' << b << ' ' << c << endl;

    dq = a * xq + b * yq + c;
    dq = dq / abs(dq);
    d = a * x[0] + b * y[0] + c;
    dmax = d * dq;
    //cout << dmax << endl;
    imax = 1;
    for(int i = 1; i < n; ++i)
    {
        d = a * x[i] + b * y[i] + c;
        //cout << d * dq << endl;
        if(d * dq == dmax)
        {
            if(abs(A * x[i] + B * y[i] + C) < abs(A * x[imax - 1] + B * y[imax - 1] + C))
            {
                //cout << abs(A * x[i] + B * y[i] + C) << " " << abs(A * x[imax] + B * y[imax] + C
                    ) << endl;
                imax = i + 1;
                dmax = d * dq;
            }
            else if(abs(A * x[i] + B * y[i] + C) == abs(A * x[imax - 1] + B * y[imax - 1] + C))
            {
                //cout << i << " " << imax << endl;
                //cout << abs(A * x[0] + B * y[0] + C) << " " << abs(A * x[1] + B * y[1] + C) <<
                    endl;
                imax = -1;
            }
        }
        if(d * dq > dmax)
        {
            imax = i + 1;
            dmax = d * dq;
        }

    }
    cout << imax /* << " " << (int)(x[1] > x[0]) + 1 */ << endl;
    return 0;
}
```

Task F (—)