Олимпиада СПбГУ по информатике 2018/19 учебного года

Быков Давид Вадимович

A	B	C	D	E	F	Sum
100	100	100	60	12	0	372

Task A (100)

```
n = int(input())
m = int(input())

if n > m:
    print('No')
else:
    if n == m:
        print('Yes')
    else:
        while n <= m:
            if n == m:
                print('Yes')
                exit()
            n *= 2
        print('No')
```

Task B (100)

```python
n = int(input())
a = input()
for i in range(1, n):
    if a[i - 1] == 'o' and a[i] == 'r' or a[i - 1] == 'r' and a[i] == 'o':
        print('Yes')
        exit()
for i in range(2, n):
    if a[i - 2] == 'o' and a[i] == 'r':
        print('Yes')
        exit()
print('No')
```

Task C (100)

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

vector<vector<int> > g;
vector<int> visited;
vector<int> su;

void dfs(int v){
    visited[v] = 1;
    int s = 0;
    for (int i = 0; i <  g[v].size(); i++)
        if (visited[g[v][i]] == 0){
            dfs(g[v][i]);
            s += su[g[v][i]];
        }
    if (s == 0){
        su[v] = 1;
    }
    else{
        su[v] = s + 1;
    }
}

int main(){
    int n;
    cin >> n;
    if (n == 1){
        cout << 1;
        return 0;
    }

    for(int i = 0; i < n; i++)
        g.push_back({});

    for(int i = 0; i < n - 1; i++){
        int p, q;
        cin >> p >> q;
        g[p - 1].push_back(q - 1);
        g[q - 1].push_back(p - 1);
    }

    for(int i = 0; i < n; i++){
        visited.push_back(0);
        su.push_back(0);
    }

    dfs(0);
    int ans = -1;
    for (int j = 0; j < g[0].size(); j++)
        ans = max(ans, su[g[0][j]]);
    cout << ans + 1<<  '␣';

    for(int i = 1; i < n; i++)
        if (su[i] == 1){
            cout << n <<  '␣';
        }
        else{
            ans = -1;
            if (g[i].size() == 1){
                    cout << su[g[i][0]] + 1 << '␣';
            }
            else{
                int maxs = -1, ans = -1;
                for (int j = 0; j < g[i].size(); j++)
                    maxs = max(maxs, su[g[i][j]]);
                for (int j = 0; j < g[i].size(); j++){
                    if (su[g[i][j]] == maxs){
                        ans = max(ans, n - su[i]);
                    }
                    else{
                        ans = max(ans, su[g[i][j]]);
                    }
                }
            cout << ans + 1 << '␣';
            }
        }
    return 0;
}
```

Task D (60)

```
test = input()
f1 = 'i'
f2 = 'qr'
if test == 'split':
    t, n, p = map(int, input().split())
    if n == 3:
        a = [input() for _ in range(t)]
        for i in a:
            print(i[0] + i[1] + i[3] + i[4] + i[6] + i[7] + 'a', i[1] + i[2] + i[4] + i[5] + i[7]
                 + i[8] + 'b', i[0] + i[2] + i[3] + i[5] + i[6] + i[8] + 'c')
    elif n == 5:
        a = [input() for _ in range(t)]
        for i in a:
            print(i[0] + i[1] + i[3] + i[4] + i[6] + i[7] + 'a', i[0] + i[1] + i[3] + i[4] + i[6]
                 + i[7] + 'a', i[1] + i[2] + i[4] + i[5] + i[7] + i[8] + 'b', i[1] + i[2] + i[4] +
                 i[5] + i[7] + i[8] + 'b', i[0] + i[2] + i[3] + i[5] + i[6] + i[8] + 'c')
    else:
        ans = ('a' * p + '␣') * n
        for i in range(t):
            print(ans[:-1])
elif test == 'merge':
    t, n, p = map(int, input().split())
    if n == 3:
        for req in range(t):
            a, b = input().split()
            if a[-1] > b[-1]:
                a, b = b, a
            if a[-1] == 'a' and b[-1] == 'b':
                print(a[0] + a[1] + b[1] + a[2] + a[3] + b[3] + a[4] + a[5] + b[5])
            if a[-1] == 'b' and b[-1] == 'c':
                print(b[0] + a[0] + a[1] + b[2] + a[2] + a[3] + b[4] + a[4] + a[5])
            if a[-1] == 'a' and b[-1] == 'c':
                print(a[0] + a[1] + b[1] + a[2] + a[3] + b[3] + a[4] + a[5] + b[5])
    elif n == 5:
        for req in range(t):
            a, b, c = input().split()
            if a == b:
                b = c
            if a[-1] > b[-1]:
                a, b = b, a
            if a[-1] == 'a' and b[-1] == 'b':
                print(a[0] + a[1] + b[1] + a[2] + a[3] + b[3] + a[4] + a[5] + b[5])
            if a[-1] == 'b' and b[-1] == 'c':
                print(b[0] + a[0] + a[1] + b[2] + a[2] + a[3] + b[4] + a[4] + a[5])
            if a[-1] == 'a' and b[-1] == 'c':
                print(a[0] + a[1] + b[1] + a[2] + a[3] + b[3] + a[4] + a[5] + b[5])

    else:
        for i in range(t):
            print('abc' * 3)
```

Task E (12)

```
def solve(x1, y1, x2, y2):
    return (x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)

n = int(input())
if n == 2:
    x1, y1, = map(int, input().split())
    x2, y2 = map(int, input().split())
    xp, yp = map(int, input().split())
    xq, yq = map(int, input().split())
    if yp != yq:
        print(1)
        x = (xq - xp) * 10000
        y = (yq - yp) * 10000
        #if solve(x2, y2, x, y) < solve(x1, y1, x, y):
            #print(1)
        #else:
            #print(2)
    if x1 == x2 and y2 + y1 == 0:
        print(-1)
    elif xp < xq:
        if x1 < x2:
            print(2)
        elif x1 > x2:
            print(1)
        else:
            if abs(y1 - yp) < abs(y2 - yp):
                print(1)
            else:
                print(2)
    elif xp > xq:
        if x1 < x2:
            print(1)
        elif x1 > x2:
            print(2)
        else:
            if abs(y1 - yp) < abs(y2 - yp):
                print(1)
            else:
                print(2)
```

Task F (—)