

Олимпиада СПбГУ по информатике 2018/19 учебного года

Радивончик Дмитрий Ильич

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

Task A (100)

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <string>
#include <algorithm>

using namespace std;

int main()
{
    int n, m;
    cin >> n >> m;
    if (n == m)
    {
        printf("Yes");
        return 0;
    }
    if (n > m)
    {
        printf("No");
        return 0;
    }
    while (1)
    {
        n *= 2;
        if (n > m)
        {
            printf("No");
            return 0;
        }
        if (n == m)
        {
            printf("Yes");
            return 0;
        }
    }
    return 0;
}
```

Task B (100)

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <string>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    cin >> n;
    string s;
    cin >> s;
    if(n <= 1)
    {
        printf("No");
        return 0;
    }
    if((s[0] == 'o' && s[1] == 'r') || (s[n - 2] == 'r' && s[n - 1] == 'o') || (s[n - 2] == 'o' &&
        s[n - 1] == 'r'))
    {
        printf("Yes");
        return 0;
    }
    if(n > 2)
    {
        if((s[0] == 'o' && s[2] == 'r') || (s[n - 3] == 'r' && s[n - 2] == 'o'))
        {
            printf("Yes");
            return 0;
        }
    }
    for(int i = 1; i < n - 2; i++)
    {
        if(s[i] == 'o')
        {
            if(s[i - 1] == 'r' || s[i + 1] == 'r' || s[i + 2] == 'r')
            {
                printf("Yes");
                return 0;
            }
        }
    }
    printf("No");
    return 0;
}
```

Task C (100)

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <string>
#include <algorithm>
#include <vector>

using namespace std;

vector<int> rb[100000];
vector<int> rbc[100000];
int n;

int find_reb(int cur, int tar)
{
    for(int i = 0; i < rb[cur].size(); i++)
    {
        if(rb[cur][i] == tar)
            return i;
    }
    return -1;
}

int count_reb(int cur, int prev)
{
    int ant, ans = 0;
    for(int i = 0; i < rb[cur].size(); i++)
    {
        if(rb[cur][i] == prev)
            continue;
        ant = count_reb(rb[cur][i], cur);
        //reb[rb[cur][i]][cur] = ant;
        rbc[cur][i] = ant;
        //reb[cur][rb[cur][i]] = n - ant;
        rbc[rb[cur][i]][find_reb(rb[cur][i], cur)] = n - ant;
        ans += ant;
    }
    return ans + 1;
}

int main()
{
    int tmp1, tmp2;
    cin >> n;

    for(int i = 0; i < n - 1; i++)
    {
        cin >> tmp1 >> tmp2;
        rb[tmp1 - 1].push_back(tmp2 - 1);
        rbc[tmp1 - 1].push_back(-1);
        rb[tmp2 - 1].push_back(tmp1 - 1);
        rbc[tmp2 - 1].push_back(-1);
    }
    count_reb(0, -1);
    for(int i = 0; i < n; i++)
    {
        int max = 0;
        for(int j = 0; j < rb[i].size(); j++)
        {
            if(rbc[i][j] > max)
                max = rbc[i][j];
        }
        printf("%d_", max + 1);
    }
    return 0;
}
```

Task D (—)

Task E (100)

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <string>
#include <algorithm>
#include <vector>

using namespace std;

long long coord[100000][2];

long long absol(long long a)
{
    if(a < 0)
        return -a;
    return a;
}

int main()
{
    int n;
    cin >> n;
    long long max_pr, x, y, x1, y1, x3, y3, prom, h, min_h;
    int num;
    int flag = 0;
    for(int i = 0; i < n; i++)
    {
        cin >> x >> y;
        coord[i][0] = x;
        coord[i][1] = y;
    }
    cin >> x1 >> y1 >> x3 >> y3;
    //printf("x1=%lld, y1=%lld, x3=%lld, y3=%lld\n", x1, y1, x3, y3);
    max_pr = (x3 - x1) * coord[0][0] + (y3 - y1) * coord[0][1];
    min_h = absol(coord[0][0] * (y3 - y1) - coord[0][1] * (x3 - x1) - x1 * (y3 - y1) + y1 * (x3 - x1));
    num = 1;
    //printf("proek=%lld, h=%lld, h bez modulya=%lld\n", max_pr, min_h, coord[0][0] * (y3 - y1) - coord[0][1] * (x3 - x1) - x1 * (y3 - y1) + y1 * (x3 - x1));
    for(int i = 1; i < n; i++)
    {
        prom = (x3 - x1) * coord[i][0] + (y3 - y1) * coord[i][1];
        h = absol(coord[i][0] * (y3 - y1) - coord[i][1] * (x3 - x1) - x1 * (y3 - y1) + y1 * (x3 - x1));
        //printf("proek=%lld, h=%lld, h bez modulya=%lld\n", prom, h, coord[i][0] * (y3 - y1) - coord[i][1] * (x3 - x1) - x1 * (y3 - y1) + y1 * (x3 - x1));
        if(prom == max_pr)
        {
            if(h == min_h)
                flag = 1;
            if(h < min_h)
            {
                min_h = h;
                num = i + 1;
                flag = 0;
            }
        }
        if(prom > max_pr)
        {
            max_pr = prom;
            min_h = h;
            num = i + 1;
            flag = 0;
        }
    }
    if(flag == 1)
        printf("-1");
    else
```

```
        printf("%d", num);  
    return 0;  
}
```

Task F (—)