

Олимпиада СПбГУ по информатике 2018/19 учебного года

Якупов Ильгиз Махмутович

A	B	C	D	E	F	Sum
100	100	100	60	0	7	367

Task A (100)

```
#include <ext/pb_ds/assoc_container.hpp>
#include <unordered_map>
#include <algorithm>
#include <iostream>
#include <vector>
#include <deque>
#include <cmath>
#include <map>
#include <set>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
//gg_hash_table<int, int>;
//cc_hash_table<int, int>;

#define deb(x) for (auto &i: x) cout << i << '\n';
#define sp << ' ' <<
#define nl << '\n'
#define sq << ' '

void solve();

int main() {
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);
    solve();
}

int n, m;

void solve() {
    cin >> n >> m;

    if (m % n != 0) {
        cout << "No";
        return;
    }

    int d = m / n;

    while (d > 0 && d % 2 == 0) {
        d /= 2;
    }
```

```
    if (d > 1) {  
        cout << "No";  
    }  
  
    else {  
        cout << "Yes";  
    }  
}
```

Task B (100)

```
#include <ext/pb_ds/assoc_container.hpp>
#include <unordered_map>
#include <algorithm>
#include <iostream>
#include <vector>
#include <deque>
#include <cmath>
#include <map>
#include <set>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
//gg_hash_table<int, int>;
//cc_hash_table<int, int>;

#define deb(x) for (auto &i: x) cout << i << '\n';
#define sp << ' ' <<
#define nl << '\n'
#define sq << ' '

void solve();

int main() {
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);
    solve();
}

int n;
string s;
void solve() {
    cin >> n >> s;
    bool ans = false;

    for (int i = 0; i < n - 1; ++i) {
        if ((s[i] == 'o' && s[i + 1] == 'r') || (s[i] == 'o' && s[i + 2] == 'r') || (s[i] == 'r'
            && s[i + 1] == 'o')) {
            ans = true;
        }
    }

    cout << (ans ? "Yes" : "No");
}

}
```

Task C (100)

```
#include <ext/pb_ds/assoc_container.hpp>
#include <unordered_map>
#include <algorithm>
#include <iostream>
#include <vector>
#include <deque>
#include <cmath>
#include <map>
#include <set>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;
//gg_hash_table<int, int>;
//cc_hash_table<int, int>;

#define deb(x) for (auto &i: x) cout << i << '\n';
#define sp << ' ' <<
#define nl << '\n'
#define sq << ' '

void solve();

int main() {
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);
    solve();
}

int n;
vector<vector<pair<int, int>>>> g;
vector<pair<int, int>> b;
vector<int> c;

void dfs(int v, int p, int k) {
    for (int i = 0; i < g[v].size(); ++i) {
        if (g[v][i].first == p) continue;
        dfs(g[v][i].first, v, i);
        g[p][k].second += g[v][i].second;
    }

    g[p][k].second += 1;
    b[v] = {p, k};
}

void solve() {
    cin >> n;
    g.resize(n);
    b.resize(n);
    c.resize(n);

    if (n == 1) {
        cout << 1;
        return;
    }

    int init = 0;

    for (int i = 1, u, v; i < n; ++i) {
        cin >> u >> v;
        u--, v--;
```

```

        c[u]++, c[v]++;
        if (c[u] == 1) init = u;
        if (c[v] == 1) init = v;
        g[u].push_back({v, 0});
        g[v].push_back({u, 0});
    }

    //cout << "init = " << init << endl;

    for (int i = 0; i < g[init].size(); ++i) {
        dfs(g[init][i].first, init, i);
    }

    /*
    for (int i = 0; i < n; ++i) {
        int ans = 0;
        for (auto &j: g[i]) {
            ans += j.second;
        }
        cout << ans << " ";
    }
    */

    for (int i = 0; i < n; ++i) {
        if (c[i] == 1) {
            cout << n << "┘";
            continue;
        }

        int leaf = 0;
        int sum = 0;
        int mx = (c[b[i].first] == 1 ? 0 : n - g[b[i].first][b[i].second].second);

        //cout << "initial mx = " << mx << endl;

        for (auto &v: g[i]) {
            if (c[v.first] == 1) leaf++, sum += v.second;

            else if (v.second != 0) {
                mx = max(mx, v.second);
            }
        }

        //cout << "mx = " << mx << endl;

        if (n - 1 - leaf == 0) cout << 2 << "┘";
        else cout << mx + 1 << "┘";
    }
}

```

Task D (60)

```
#include <ext/pb_ds/assoc_container.hpp>
#include <unordered_map>
#include <algorithm>
#include <iostream>
#include <vector>
#include <deque>
#include <cmath>
#include <map>
#include <set>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
//gg_hash_table<int, int>;
//cc_hash_table<int, int>;

#define deb(x) for (auto &i: x) cout << i << '\n';
#define sp << ' ' <<
#define nl << '\n'
#define sq << ' '

void solve();

int main() {/*
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);*/
    solve();
}

string op;
int t, n, p;

void split() {
    while (t--) {
        string s;
        cin >> s;
        string s1, s2, s3, s4, s5;

        if (n == 3) {
            s1 = "a" + string(1, s[8]) + s.substr(0, 5);
            s2 = "b" + s.substr(2, 6);
            s3 = "c" + s.substr(5) + s.substr(0, 2);
            cout << s1 sp s2 sp s3 nl;
        }

        if (n == 5) {
            s1 = "a" + string(1, s[8]) + s.substr(0, 5);
            s2 = "b" + s.substr(2, 6);
            s3 = "c" + s.substr(5) + s.substr(0, 2);
            s4 = "a" + string(1, s[8]) + s.substr(0, 5);
            s5 = "b" + s.substr(2, 6);
            cout << s1 sp s2 sp s3 sp s4 sp s5 nl;
        }
    }
}

void defme(string a, string b) {
    int p1 = a[0] - 'a';
    int p2 = b[0] - 'a';
    if (p1 > p2) swap(p1, p2), swap(a, b);

    if (p1 == 0) {
```

```

        cout << a.substr(2);
        if (p2 == 1) {
            cout << b.substr(4) + a[1] nl;
        } else {
            cout << b.substr(1, 4) nl;
        }
    }
    else {
        cout << b.substr(5) + a.substr(1) + b[4] nl;
    }
}

void defme(string a, string b, string c) {
    int p1 = a[0] - 'a';
    int p2 = b[0] - 'a';
    int p3 = c[0] - 'a';

    if (a == b) defme(a, c);
    else if (a == c) defme(a, b);
    else defme(a, c);
}

void merge() {
    while (t--) {
        string s1, s2, s3, s4, s5;

        if (n == 3) {
            string s1, s2;
            cin >> s1 >> s2;
            defme(s1, s2);
        }
        if (n == 5) {
            string s1, s2, s3;
            cin >> s1 >> s2 >> s3;
            defme(s1, s2, s3);
        }
    }
}

void solve() {
    cin >> op >> t >> n >> p;

    if (op == "split") split();
    else merge();
}

```

Task E (—)

Task F (7)

```
#include <ext/pb_ds/assoc_container.hpp>
#include <unordered_map>
#include <algorithm>
#include <iostream>
#include <vector>
#include <deque>
#include <cmath>
#include <map>
#include <set>

using namespace __gnu_pbds;
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
//gg_hash_table<int, int>;
//cc_hash_table<int, int>;

#define deb(x) for (auto &i: x) cout << i << '\n';
#define sp << ' ' <<
#define nl << '\n'
#define sq << ' '

void solve();

int main() {
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);
    solve();
}

int n, k, mx = 0;
vector<pair<int, int>> dd;

void rec(int i, int r, int b, int h) {

    //cout << "day " << i << " r = " << r << " b = " << b << " h = " << h << endl;

    if (i == n) {
        mx = max(h, mx);
        return;
    }

    int rd = dd[i].first - max(0, dd[i].first - r);
    r -= rd, h += rd;

    int bd = dd[i].second - max(0, dd[i].second - b);
    b -= bd, h += bd;

    rec(i + 1, r + k, b, h);
    rec(i + 1, r, k + b, h);
}

void solve() {
    cin >> n >> k;

    dd.resize(n);

    for (auto &i: dd) {
        cin >> i.first >> i.second;
    }

    rec(0, k, 0, 0);
    rec(0, 0, k, 0);
}
```

```
    cout << mx;  
}
```