

Олимпиада СПбГУ по информатике 2018/19 учебного года

Филиппов Андрей Владимирович

A	B	C	D	E	F	Sum
100	100	100	60	35	7	402

Task A (100)

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    int n, m;
    cin >> n >> m;

    if (m % n) {
        cout << "No" << endl;
        return 0;
    }
    int d = m / n;
    for (int c = 0; c < 20; ++c)
        if ((1 << c) == d) {
            cout << "Yes" << endl;
            return 0;
        }
    cout << "No" << endl;

    return 0;
}
```

Task B (100)

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n;
    cin >> n;
    string s;
    cin >> s;

    for (int i = 1; i < n; ++i) {
        if (i > 0 && (s[i - 1] == 'r' && s[i] == 'o' || s[i - 1] == 'o' && s[i] == 'r') || i > 1
            && (s[i - 2] == 'o' && s[i] == 'r')) {
            cout << "Yes" << endl;
            return 0;
        }
    }

    cout << "No" << endl;

    return 0;
}
```

Task C (100)

```
#include <bits/stdc++.h>

using namespace std;

#define sz(a) ((int)a.size())

const int MAX_N = (int)1e5;

vector<int> g[MAX_N];

int cnt[MAX_N];
int cntmx[MAX_N];

void dfs(int v, int p = -1) {
    cnt[v] = 1;
    cntmx[v] = 0;
    for (int i = 0; i < sz(g[v]); ++i) {
        int to = g[v][i];
        if (to == p)
            continue;
        dfs(to, v);
        cnt[v] += cnt[to];
        cntmx[v] = max(cntmx[v], cnt[to]);
    }
}

int main()
{
    int n;
    cin >> n;

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

    dfs(0);

    for (int i = 0; i < n; ++i) {
        int ans = max(cntmx[i], cnt[0] - cnt[i]) + 1;
        cout << ans << " ";
    }
    cout << endl;

    return 0;
}
```

Task D (60)

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string q;
    cin >> q;
    bool sp = q == "split";
    int t, n, p;
    cin >> t >> n >> p;
    while (t--) {
        if (sp) {
            string s;
            cin >> s;
            if (n == 3 && p == 7) {
                cout << "a" + s.substr(0, 6) << "␣";
                cout << "b" + s.substr(3, 6) << "␣";
                cout << "c" + s.substr(0, 3) + s.substr(6, 3) << endl;
            } else if (n == 5 && p == 7) {
                cout << "a" + s.substr(0, 6) << "␣";
                cout << "b" + s.substr(3, 6) << "␣";
                cout << "c" + s.substr(0, 3) + s.substr(6, 3) << "␣";
                cout << "a" + s.substr(0, 6) << "␣";
                cout << "b" + s.substr(3, 6) << endl;
            } else if (n == 7 && p == 4) {
                ;
            }
        } else {
            string s[7];
            int k = (n + 1) / 2;
            for (int i = 0; i < k; ++i)
                cin >> s[i];
            sort(s, s + k);
            if (n == 3 && p == 7) {
                if (s[0][0] == 'a' && s[1][0] == 'b')
                    cout << s[0].substr(1, 6) + s[1].substr(4, 3) << endl;
                else if (s[0][0] == 'a' && s[1][0] == 'c')
                    cout << s[0].substr(1, 6) + s[1].substr(4, 3) << endl;
                else if (s[0][0] == 'b' && s[1][0] == 'c')
                    cout << s[1].substr(1, 3) + s[0].substr(1, 6) << endl;
            } else if (n == 5 && p == 7) {
                k = unique(s, s + k) - s;
                if (s[0][0] == 'a' && s[1][0] == 'b')
                    cout << s[0].substr(1, 6) + s[1].substr(4, 3) << endl;
                else if (s[0][0] == 'a' && s[1][0] == 'c')
                    cout << s[0].substr(1, 6) + s[1].substr(4, 3) << endl;
                else if (s[0][0] == 'b' && s[1][0] == 'c')
                    cout << s[1].substr(1, 3) + s[0].substr(1, 6) << endl;
            } else if (n == 7 && p == 4) {
                ;
            }
        }
    }

    return 0;
}

/*
split
4 3 7
passwords
apasswo bswords cpasrds
uhaaaaaaaa
auhaaaaa baaaaaa cuhaaaa
aaaaaaaaa
aaaaaaa baaaaaa caaaaaa
plainword
aplainw binword cplaord
```

```
split
1 3 7
abcdefghi
aabcdef bdefghi cabcgghi
```

```
split
1 5 7
012345678
a012345 b345678 c012678 a012345 b345678
*/
```

Task E (35)

```
#include <bits/stdc++.h>

using namespace std;

typedef long double ld;

const ld EPS = 1e-9;

struct pt {
    ld x, y;
    int i;
};

struct line {
    ld a, b, c;
};

const int MAX_N = (int)1e5;

pt p[MAX_N];

bool cmp(pt &a, pt &b) {
    //return a.x < b.x || a.x == b.x && a.y < b.y; //EPS
    return a.x + EPS < b.x || (a.x - b.x) < EPS && a.y < b.y; //EPS
}

pt a[MAX_N];

ld sqr(ld a) {
    return a * a;
}

ld get(line &l, ld x) {
    return (-l.a * x - l.c) / l.b;
}

int main()
{
    int n;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> p[i].x >> p[i].y;
        p[i].i = i;
    }

    ld x1, y1, x2, y2;
    cin >> x1 >> y1 >> x2 >> y2;

    line l = {y1 - y2, x2 - x1, x1 * y2 - x2 * y1};
    //cout << l.a << " " << l.b << " " << l.c << endl;
    ld cosa = (ld)1e18 / sqrt(sqr(get(l, 1e18)) + sqr(1e18));
    //cout << fixed << setprecision(15) << cosa << endl;
    if (cosa < EPS)
        for (int i = 0; i < n; ++i)
            swap(p[i].x, p[i].y);
    else
        for (int i = 0; i < n; ++i) {
            ld d = abs(l.a * p[i].x + l.b * p[i].y + l.c) / sqrt(l.a * l.a + l.b * l.b);
            ld g = sqrt(sqr(p[i].y - get(l, p[i].x)) - d * d);
            ld dx = g * cosa;
            bool up = p[i].y > get(l, p[i].x);
            ld dxn = p[i].x + dx * (up ? 1 : -1);
            p[i].x = dxn / cosa;
            p[i].y = d * (up ? 1 : -1);
            //cout << p[i].x << " " << p[i].y << " " << d << " " << g << " " << dx << endl;
        }

    sort(p, p + n, cmp);
    ld x = p[n - 1].x;
```

```

int m = 0;
for (int i = 0; i < n; ++i)
    if (abs(p[i].x - x) < EPS) //EPS
        a[m++] = p[i];
int j;
for (j = 0; j < m; ++j)
    if (a[j].y > EPS) //EPS
        break;
--j;
if (j == -1)
    cout << a[j + 1].i + 1 << endl;
else if (j + 1 == m)
    cout << a[j].i + 1 << endl;
else {
    if (abs(a[j].y) + EPS < abs(a[j + 1].y)) //EPS
        cout << a[j].i + 1 << endl;
    else if (abs(a[j].y) > abs(a[j + 1].y + EPS)) //EPS
        cout << a[j + 1].i + 1 << endl;
    else
        cout << -1 << endl;
}
return 0;
}

```

Task F (7)

```
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const int MAX_N = 15;
const int MAX_K = 15;
const int MAX_A = MAX_N * MAX_K;

int dp[MAX_N + 1][MAX_A + 1][MAX_A + 1][2];

int main()
{
    int n, k;
    cin >> n >> k;

    memset(dp, 255, sizeof(dp));
    dp[0][0][0][0] = 0;
    for (int i = 0; i < n; ++i) {
        int r, b;
        cin >> r >> b;
        for (int x = 0; x <= MAX_A; ++x)
            for (int y = 0; y <= MAX_A; ++y) {
                if (dp[i][x][y][0] == -1)
                    continue;
                dp[i][x + k][y][1] = max(dp[i][x + k][y][1],
                                           dp[i][x][y][0]);
                dp[i][x][y + k][1] = max(dp[i][x][y + k][1],
                                           dp[i][x][y][0]);
            }
        for (int x = 0; x <= MAX_A; ++x)
            for (int y = 0; y <= MAX_A; ++y) {
                if (dp[i][x][y][1] == -1)
                    continue;
                dp[i + 1][max(0, x - r)][max(0, y - b)][0] =
                    max(dp[i + 1][max(0, x - r)][max(0, y - b)][0],
                        dp[i][x][y][1] + min(x, r) + min(y, b));
            }
    }

    int ans = 0;
    for (int x = 0; x <= MAX_A; ++x)
        for (int y = 0; y <= MAX_A; ++y)
            ans = max(ans, dp[n][x][y][0]);

    cout << ans << endl;

    return 0;
}
```