

Олимпиада СПбГУ по информатике 2018/19 учебного года

Наумов Иван Леонидович

A	B	C	D	E	F	Sum
100	100	100	60	21	24	405

Task A (100)

```
#pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
```

```

        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 3e5 + 5, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

int n, m;

int main() {
    scanf("%d_%d", &n, &m);
    while (n < m) {
        n *= 2;
    }
    if (n == m) {
        printf("Yes\n");
    } else {
        printf("No\n");
    }
    return 0;
}

```

Task B (100)

```
#pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 3e5 + 5, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

int n;
char s[MAXN];

int main() {
    scanf("%d\n", &n);
    scanf("%s", s);
```

```

for (int i = 0; i + 1 < n; ++i) {
    if (s[i] == 'o' && s[i + 1] == 'r' ||
        s[i] == 'r' && s[i + 1] == 'o') {
        printf("Yes\n");
        return 0;
    }
    if (i + 2 < n && s[i] == 'o' && s[i + 2] == 'r') {
        printf("Yes\n");
        return 0;
    }
}
printf("No\n");
return 0;
}

```

Task C (100)

```
#pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 1e5 + 5, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

int n, sz[MAXN], ans[MAXN];
vector<int> g[MAXN];

void calc_sz(int v, int p) {
    sz[v] = 1;
    for (auto &to : g[v]) {
```

```

        if (to != p) {
            calc_sz(to, v);
            sz[v] += sz[to];
        }
    }
    ans[v] = n - sz[v];
}

int dfs(int v, int p) {
    for (auto &to : g[v]) {
        if (to != p) {
            ans[v] = max(ans[v], sz[to]);
        }
    }
    for (auto &to : g[v]) {
        if (to != p) {
            dfs(to, v);
        }
    }
}

int main() {
    int v, u, maxSubTree;
    scanf("%d", &n);
    for (int i = 0; i < n - 1; ++i) {
        scanf("%d_%d", &v, &u);
        --v, --u;
        g[v].emplace_back(u);
        g[u].emplace_back(v);
    }
    calc_sz(0, -1);
    dfs(0, -1);
    for (int i = 0; i < n; ++i) {
        printf("%d_", ans[i] + 1);
    }
    return 0;
}

```

Task D (60)

```
//#pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 1e5 + 1, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

int t, n, p;
string s;

char inc(int c, int d) {
    c += d;
    if (c > 'z') c -= 26;
```

```

    return (char)c;
}
char dec(int c, int d) {
    c -= d;
    if (c < 'a') c += 26;
    return (char)c;
}
int distance(char c1, char c2) {
    int d = c2 - c1;
    if (abs(d) > 10) {
        if (d < 0) d += 26;
        else d -= 26;
    }
    return d;
}
int longestPrefix(const string &s1, const string &s2) {
    int i = 0;
    while (s1[i] == dec(s2[i], 1)) {
        ++i;
    }
    return i;
}

int main() {
    cin >> s;
    cin >> t >> n >> p;
    if (n == 3) {
        if (s[0] == 's') {
            while (t--) {
                cin >> s;
                for (int i = 0; i < 7; ++i) cout << s[i];
                cout << '\n';
                for (int i = 0; i < 5; ++i) cout << inc(s[i], 1);
                for (int i = 7; i < 9; ++i) cout << inc(s[i], 1);
                cout << '\n';
                for (int i = 0; i < 3; ++i) cout << inc(s[i], 3);
                for (int i = 5; i < 9; ++i) cout << inc(s[i], 3);
                cout << endl;
            }
        } else {
            string s1, s2;
            while (t--) {
                cin >> s1 >> s2;
                int d = distance(s1[0], s2[0]);
                if (d < 0) {
                    swap(s1, s2);
                    d = -d;
                }
                if (d == 1) {
                    for (int i = 0; i < 7; ++i) cout << s1[i];
                    for (int i = 5; i < 7; ++i) cout << dec(s2[i], 1);
                } else if (d == 2) {
                    for (int i = 0; i < 5; ++i) cout << dec(s1[i], 1);
                    for (int i = 3; i < 7; ++i) cout << dec(s2[i], 3);
                } else {
                    for (int i = 0; i < 7; ++i) cout << s1[i];
                    for (int i = 5; i < 7; ++i) cout << dec(s2[i], 3);
                }
                cout << endl;
            }
        }
    } else if (n == 5) {
        if (s[0] == 's') {
            while (t--) {
                cin >> s;
                for (int i = 0; i < 7; ++i) cout << s[i];
                cout << '\n';
                for (int i = 0; i < 6; ++i) cout << inc(s[i], 1);
                for (int i = 8; i < 9; ++i) cout << inc(s[i], 1);
                cout << '\n';
                for (int i = 0; i < 5; ++i) cout << inc(s[i], 3);
                for (int i = 7; i < 9; ++i) cout << inc(s[i], 3);
                cout << '\n';
            }
        }
    }
}

```



```

        for (int i = 0; i < 4; ++i) cout << inc(s[i], 6);
        for (int i = 6; i < 9; ++i) cout << inc(s[i], 6);
        cout << '┘';
        for (int i = 0; i < 3; ++i) cout << inc(s[i], 10);
        for (int i = 5; i < 9; ++i) cout << inc(s[i], 10);
        cout << endl;
    }
} else {
    while (t--) {
        string s1, s2, s3;
        cin >> s1 >> s2 >> s3;
        if (distance(s1[0], s2[0]) < 0) swap(s1, s2);
        if (distance(s1[0], s3[0]) < 0) swap(s1, s3);
        if (distance(s2[0], s3[0]) < 0) swap(s2, s3);
        int d1 = distance(s1[0], s2[0]), d2 = distance(s1[0], s3[0]);
        /*cout << "d1 = " << d1 << endl;
        cout << "d2 = " << d2 << endl;
        cout << "s1 = " << s1 << endl;
        cout << "s2 = " << s2 << endl;
        cout << "s3 = " << s3 << endl;*/
        if (d1 == 1 && d2 == 3) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s3[i], 3);
        } else if (d1 == 1 && d2 == 6) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s3[i], 6);
        } else if (d1 == 1 && d2 == 10) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s3[i], 10);
        } else if (d1 == 3 && d2 == 6) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s2[i], 3);
        } else if (d1 == 3 && d2 == 10) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s2[i], 3);
        } else if (d1 == 6 && d2 == 10) {
            for (int i = 0; i < 7; ++i) cout << s1[i];
            for (int i = 5; i < 7; ++i) cout << dec(s2[i], 6);
        } else if (d1 == 2 && d2 == 5) {
            for (int i = 0; i < 6; ++i) cout << dec(s1[i], 1);
            cout << dec(s3[4], 6);
            for (int i = 5; i < 7; ++i) cout << dec(s2[i], 3);
        } else if (d1 == 2 && d2 == 9) {
            for (int i = 0; i < 6; ++i) cout << dec(s1[i], 1);
            for (int i = 4; i < 7; ++i) cout << dec(s3[i], 10);
        } else if (d1 == 5 && d2 == 9) {
            for (int i = 0; i < 6; ++i) cout << dec(s1[i], 1);
            for (int i = 4; i < 7; ++i) cout << dec(s3[i], 10);
        } else {
            for (int i = 0; i < 5; ++i) cout << dec(s1[i], 3);
            for (int i = 3; i < 7; ++i) cout << dec(s3[i], 10);
        }
        cout << endl;
    }
}
} else {
    cout << "No" << endl;
}
return 0;
}
/*
passwor qbttxpt sdvvzgv vgyyxjy zkcybnc
*/

```

Task E (21)

```

// #pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 1e5 + 1, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

bool eq(double a, double b) {
    return fabs(a - b) < eps;
}

bool ls(double a, double b) {
    return a < b && !eq(a, b);
}

bool lseq(double a, double b) {
    return a < b || eq(a, b);
}
```

```

}
bool gr(double a, double b) {
    return a > b && !eq(a, b);
}
bool greq(double a, double b) {
    return a > b || eq(a, b);
}
int sign(double a) {
    if (a < 0) return -1;
    else return 1;
}
struct Point {
    double x, y;

    Point() : x(), y() {}
    Point(double x, double y) : x(x), y(y) {}

    void scan() {
        scanf("%lf_%lf", &x, &y);
    }
    void print() {
        if (fabs(x) < eps) x = 0;
        if (fabs(y) < eps) y = 0;
        printf("%.12lf,_%%.12lf\n", x, y);
    }

    Point operator * (const double &k) const {
        return Point(x * k, y * k);
    }
    Point operator / (const double &k) const {
        return Point(x / k, y / k);
    }
    Point operator + (const Point &a) const {
        return Point(x + a.x, y + a.y);
    }
    Point operator - (const Point &a) const {
        return Point(x - a.x, y - a.y);
    }
    double operator % (const Point &a) const {
        return x * a.x + y * a.y;
    }
    double operator * (const Point &a) const {
        return x * a.y - a.x * y;
    }
    double sqrLen() const {
        return *this % *this;
    }
    double len() const {
        return sqrt(sqrLen());
    }
    Point norm() const {
        return *this / len();
    }
    Point ort() const {
        return Point(-y, x);
    }
};

double dist(const Point &a, const Point &b) {
    return (a - b).len();
}

struct Line {
    double A, B, C;

    Line() : A(), B(), C() {}
    Line(double A, double B, double C) : A(A), B(B), C(C) {}
    Line(const Point &a, const Point &b) : A(b.y - a.y), B(a.x - b.x), C(b.x * a.y - a.x * b.y) {}

```

```

void print() const {
    printf("%.12lf_%.12lf_%.12lf\n", A, B, C);
}

double val(const Point &a) const {
    return A * a.x + B * a.y + C;
}
bool operator == (const Line &l) const {
    return eq(A / l.A, B / l.B) && eq(B / l.B, C / l.C);
}
};

double dist(const Point &a, const Line &l) {
    return fabs(l.A * a.x + l.B * a.y + l.C) / sqrt(l.A * l.A + l.B * l.B);
}

bool Intersect(const Line &l1, const Line &l2) {
    return !eq(l1.A * l2.B, l2.A * l1.B);
}
Point GetIntersect(const Line &l1, const Line &l2) {
    return Point(l1.B * l2.C - l2.B * l1.C, l2.A * l1.C - l1.A * l2.C) / (l1.A * l2.B - l2.A * l1.
        B);
}

int n, ind[MAXN];
Point p, q, a, b, city[MAXN];

int main() {
    scanf("%d", &n);
    if (n == 2) {
        scanf("%lf_%lf", &a.x, &a.y);
        scanf("%lf_%lf", &b.x, &b.y);
        scanf("%lf_%lf", &p.x, &p.y);
        scanf("%lf_%lf", &q.x, &q.y);
        Line l(p, q), l1((a + b) / 2, (a + b) / 2 + (b - a).ort());
        if (!Intersect(l, l1)) {
            if (eq(dist(a, l), dist(b, l))) {
                printf("-1\n");
            } else if (ls(dist(a, l), dist(b, l))) {
                printf("1\n");
            } else {
                printf("2\n");
            }
            return 0;
        } else {
            Point c = GetIntersect(l, l1);
            Point d = c + (q - p);
            if (sign(l1.val(d)) == sign(l1.val(a))) {
                printf("1\n");
            } else {
                printf("2\n");
            }
            return 0;
        }
    }
    for (int i = 0; i < n; ++i) {
        city[i].scan();
    }
    p.scan(), q.scan();
    Point z(2e18, 0);
    for (int i = 0; i < n; ++i) ind[i] = i;
    sort(ind, ind + n, [&](int ind1, int ind2) {
        return ls(dist(z, city[ind1]), dist(z, city[ind2]));
    });
    if (eq(dist(z, city[ind[0]]), dist(z, city[ind[1]]))) {
        printf("-1\n");
    } else {

```

```
        printf("%d\n", ind[0]);
    }
    return 0;
}

/*
2
0 100
0 -99
0 0
100 0
*/
```

Task F (24)

```
#pragma GCC optimize("Ofast")
#include <bits/stdc++.h>
#define mp make_pair
using namespace std;
typedef long long ll;
typedef unsigned int uint;
typedef unsigned long long ull;

template<class ForwardIterator>
void showArray(ForwardIterator first, ForwardIterator last, const string& arrName) {
    cout << arrName << ":[ ";
    if (first != last) {
        cout << *first++;
    }
    while (first != last) {
        cout << ", " << *first++;
    }
    cout << "]" << endl;
}

int readInt() {
    int res = 0;
    char c = getchar(), sign = true;
    if (c == '-') {
        sign = false;
        c = getchar();
    }
    while (c != '-' && c != '\n') {
        res = res * 10 + c - '0';
        c = getchar();
    }
    if (!sign)
        res = -res;
    return res;
}

void printInt(int n) {
    if (!n) {
        putchar('0');
        return;
    }
    if (n < 0) {
        putchar('-');
        n = -n;
    }
    char st[10], *p = st + 10;
    while (n) {
        *--p = (n % 10) + '0';
        n /= 10;
    }
    while (p != st + 10) {
        putchar(*p++);
    }
}

const bool debug = true;
const int MAXN = 1e5 + 5, INF = 2e9;
const double pi = acos(-1), eps = 1e-9;

int n, k, r[MAXN], b[MAXN], ans;
bool dp[31][30 * 30 + 1][30 * 30 + 1];

int main() {
    ans = 0;
    scanf("%d_%d", &n, &k);
```

```

int maxCnt = n * k;
for (int i = 0; i < n; ++i) {
    scanf("%d%d", &r[i], &b[i]);
}
dp[0][0][0] = 1;
for (int i = 0; i < n; ++i) {
    for (int j = 0; j <= maxCnt; ++j) {
        for (int t = 0; t <= maxCnt; ++t) {
            if (dp[i][j][t]) {
                ans = max(ans, i * k - j - t);
                /*dp[i + 1][j + k][t] = 1;
                dp[i + 1][j][t + k] = 1;
                dp[i + 1][j + k - min(r[i], j + k)][t] = 1;
                dp[i + 1][j + k - min(r[i], j + k)][t - min(b[i], t)] = 1;
                dp[i + 1][j][t + k - min(b[i], t + k)] = 1;*/
                //bake red
                dp[i + 1][j + k][t] = 1;
                dp[i + 1][j + k - min(r[i], j + k)][t] = 1;
                dp[i + 1][j + k][t - min(b[i], t)] = 1;
                dp[i + 1][j + k - min(r[i], j + k)][t - min(b[i], t)] = 1;
                //bake blue
                dp[i + 1][j][t + k] = 1;
                dp[i + 1][j - min(r[i], j)][t + k] = 1;
                dp[i + 1][j][t + k - min(b[i], t + k)] = 1;
                dp[i + 1][j - min(r[i], j)][t + k - min(b[i], t + k)] = 1;
            }
        }
    }
}
for (int j = 0; j <= maxCnt; ++j) {
    for (int t = 0; t <= maxCnt; ++t) {
        if (dp[n][j][t])
            ans = max(ans, maxCnt - j - t);
    }
}
printf("%d\n", ans);
return 0;
}

```