Олимпиада СПбГУ по информатике 2018/19 учебного года

Акилов Денис Сергеевич

A	B	C	D	E	F	Sum
100	100	100	60	100	7	467

Task A (100)

```cpp
#include <iostream>

using namespace std;

int main() {
	int n, m;
	cin >> n >> m;
	while (n < m) {
		n *= 2;
	}
	cout << (n == m ? "Yes" : "No") << endl;
	//system("pause");
	return 0;
}
```

Task B (100)

```cpp
#include <iostream>
#include <algorithm>
#include <string>

using namespace std;

bool check(string s) {
	for (int i = 0; i < s.size(); ++i) {
		if (s[i] == 'o') {
			for (int j = max(i - 1, 0); j <= min(i + 2, (int)s.size() - 1); ++j) {
				if (s[j] == 'r') {
					return true;
				}
			}
		}
	}
	return false;
}

int main() {
	ios_base::sync_with_stdio(false);
	cin.tie(NULL);
	cout.tie(NULL);
	int n;
	cin >> n;
	string s;
	cin >> s;
	cout << (check(s) ? "Yes" : "No") << endl;
	//system("pause");
	return 0;
}
```

Task C (100)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int dfs(vector<vector<int>> &g, vector<int> &dp, vector<int> &prev, int v, int last = -1) {
	prev[v] = last;
	dp[v] = 1;
	for (auto u : g[v]) {
		if (u != last) {
			dp[v] += dfs(g, dp, prev, u, v);
		}
	}
	return dp[v];
}

int getAns(vector<vector<int>> &g, vector<int> &dp, vector<int> &prev, int v) {
	int res = g.size() - dp[v];
	for (auto u : g[v]) {
		if (u != prev[v]) {
			res = max(res, dp[u]);
		}
	}
	return res;
}

int main() {
	int n;
	cin >> n;
	vector<vector<int>> g(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);
	}
	int root = 0;
	vector<int> dp(n);
	vector<int> prev(n);
	dfs(g, dp, prev, root);
	for (int v = 0; v < n; ++v) {
		cout << getAns(g, dp, prev, v) + 1 << "␣";
	}
	cout << endl;
	//system("pause");
	return 0;
}
```

Task D (60)

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void Solve1_1(int t, int n, int p) {
        vector<vector<int>> arr(n);
        arr[0] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[1] = { 0, 4, 5, 6, 7, 8, 9 };
        arr[2] = { 0, 7, 8, 9, 1, 2, 3 };
        while (t--) {
                string password;
                cin >> password;
                //vector<vector<char>> ans(n, vector<char>(p));
                for (int i = 0; i < n; ++i) {
                        //ans[i][0] = 'a' + i;
                        cout << (char)('a' + i);
                        for (int j = 1; j < p; ++j) {
                                //ans[i][j] = password[arr[i][j] - 1];
                                cout << password[arr[i][j] - 1];
                        }
                        cout << " ";
                }
                cout << endl;
        }
}

void Solve1_2(int t, int n, int p) {
        vector<vector<int>> arr(n);
        arr[0] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[1] = { 0, 4, 5, 6, 7, 8, 9 };
        arr[2] = { 0, 7, 8, 9, 1, 2, 3 };
        while (t--) {
                vector<char> password(9);
                vector<vector<char>> ans((n + 1) / 2, vector<char>(p));
                for (int i = 0; i < ans.size(); ++i) {
                        for (int j = 0; j < p; ++j) {
                                cin >> ans[i][j];
                        }
                        int k = ans[i][0] - 'a';
                        for (int j = 1; j < p; ++j) {
                                password[arr[k][j] - 1] = ans[i][j];
                        }
                }
                for (int i = 0; i < 9; ++i) {
                        cout << password[i];
                }
                cout << endl;
        }
}

void Solve1(string s, int t, int n, int p) {
        if (s == "split") {
                Solve1_1(t, n, p);
        }
        else {
                Solve1_2(t, n, p);
        }
}

void Solve2_1(int t, int n, int p) {
        vector<vector<int>> arr(n);
        arr[0] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[1] = { 0, 4, 5, 6, 7, 8, 9 };
        arr[2] = { 0, 7, 8, 9, 1, 2, 3 };
        arr[3] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[4] = { 0, 4, 5, 6, 7, 8, 9 };
        while (t--) {
```

```
                string password;
                cin >> password;
                //vector<vector<char>> ans(n, vector<char>(p));
                for (int i = 0; i < n; ++i) {
                        //ans[i][0] = 'a' + i;
                        cout << (char)('a' + i);
                        for (int j = 1; j < p; ++j) {
                                //ans[i][j] = password[arr[i][j] - 1];
                                cout << password[arr[i][j] - 1];
                        }
                        cout << " ";
                }
                cout << endl;
        }
}

void Solve2_2(int t, int n, int p) {
        vector<vector<int>> arr(n);
        arr[0] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[1] = { 0, 4, 5, 6, 7, 8, 9 };
        arr[2] = { 0, 7, 8, 9, 1, 2, 3 };
        arr[3] = { 0, 1, 2, 3, 4, 5, 6 };
        arr[4] = { 0, 4, 5, 6, 7, 8, 9 };
        while (t--) {
                vector<char> password(9);
                vector<vector<char>> ans((n + 1) / 2, vector<char>(p));
                for (int i = 0; i < ans.size(); ++i) {
                        for (int j = 0; j < p; ++j) {
                                cin >> ans[i][j];
                        }
                        int k = ans[i][0] - 'a';
                        for (int j = 1; j < p; ++j) {
                                password[arr[k][j] - 1] = ans[i][j];
                        }
                }
                for (int i = 0; i < 9; ++i) {
                        cout << password[i];
                }
                cout << endl;
        }
}

void Solve2(string s, int t, int n, int p) {
        if (s == "split") {
                Solve2_1(t, n, p);
        }
        else {
                Solve2_2(t, n, p);
        }
}

int main() {
        string s;
        int t, n, p;
        cin >> s >> t >> n >> p;
        if (n == 3) {
                Solve1(s, t, n, p);
        }
        else if (n == 5) {
                Solve2(s, t, n, p);
        }
        //system("pause");
        return 0;
}
/*
split
4 3 7
passwords
apasswo bswords crdspas
uhaaaaaaa
auhaaaa baaaaaa caaauha
aaaaaaaaa
aaaaaaa baaaaaa caaaaaa
plainword
```

```
aplainw binword cordpla
*/
```

Task E (100)

```cpp
#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>

using namespace std;

#define all(v) v.begin(), v.end()

struct Point {
	long double x, y;

	Point(long double _x = 0, long double _y = 0) : x(_x), y(_y) {}
};

istream& operator>>(istream &in, Point &A) {
	in >> A.x >> A.y;
	A.x *= 2;
	A.y *= 2;
	return in;
}

bool operator!=(Point A, Point B) {
	return A.x != B.x || A.y != B.y;
}

Point mid(Point A, Point B) {
	return { (A.x + B.x) / 2, (A.y + B.y) / 2 };
}

struct Vector {
	long double x, y;

	Vector(long double _x = 0, long double _y = 0) : x(_x), y(_y) {}

	Vector(Point A, Point B) : x(B.x - A.x), y(B.y - A.y) {}
};

Point operator+(Point A, Vector a) {
	return { A.x + a.x, A.y + a.y };
}

Vector normal(Vector a) {
	return { -a.y, a.x };
}

struct Line {
	long double a, b, c;

	Line(long double _a = 0, long double _b = 0, long double _c = 0) : a(_a), b(_b), c(_c) {}

	Line(Point A, Point B) {
		if (A.y == B.y) {
			a = 0;
			b = 1;
			c = -A.y;
		}
		else {
			a = (A.y - B.y);
			b = (B.x - A.x);
			c = -A.x * a - b * A.y;
		}
		if (a < 0) {
			a = -a;
			b = -b;
			c = -c;
		}
	}
```

```
        long double check(Point A) {
                return a * A.x + b * A.y + c;
        }
};

int sign(long double d) {
        return d < 0 ? -1 : 1;
}

bool ok(Line l, Point P, Point Q, Point A) {
        long double d1 = l.check(P);
        long double d2 = l.check(Q);
        long double d = l.check(A);
        if (d1 == d2) {
                if (d1 == 0) {
                        return d > 0;
                }
                else {
                        return sign(d1) == sign(d);
                }
        }
        else if (sign(d1) != sign(d2)) {
                return sign(d1) != sign(d);
        }
        else if (abs(d1) < abs(d2)){
                return sign(d1) == sign(d);
        }
        else {
                return sign(d1) != sign(d);
        }
}

int main() {
        ios_base::sync_with_stdio(false);
        cin.tie(NULL);
        cout.tie(NULL);
        int n;
        cin >> n;
        vector<Point> arr(n);
        vector<pair<Point, int>> arrWI(n);
        for (int i = 0; i < n; ++i) {
                cin >> arr[i];
                arrWI[i] = { arr[i], i };
        }
        Point P, Q;
        cin >> P >> Q;
        if (n == 1) {
                cout << 1 << endl;
                return 0;
        }
        sort(all(arr), [=](Point A, Point B) {
                Vector a = normal(Vector(A, B));
                Line l(mid(A, B), mid(A, B) + a);
                return ok(l, P, Q, A);
        });
        Point A = arr[0], B = arr[1];
        Vector a = normal(Vector(A, B));
        Line l(mid(A, B), mid(A, B) + a);
        if (l.check(P) == 0 && l.check(Q) == 0) {
                cout << -1 << endl;
        }
        else {
                int ans = 0;
                while (arrWI[ans].first != arr[0]) {
                        ++ans;
                }
                cout << ans + 1 << endl;
        }
        //system("pause");
        return 0;
}
/*
2
0 0
```

```
1 0
0 0
1 0
*/
```

Task F (7)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int INF = 1e9;

int f(vector<pair<int, int>> &arr, int k, int mask) {
	int res = 0;
	int sR = 0, sB = 0;
	for (int i = 0; i < arr.size(); ++i) {
		int r = arr[i].first;
		int b = arr[i].second;
		if (mask & (1 << i)) {
			sR += k;
		}
		else {
			sB += k;
		}
		int _r = r;
		r = max(0, r - sR);
		sR -= _r - r;
		int _b = b;
		b = max(0, b - sB);
		sB -= _b - b;
		res += r + b;
	}
	return res;
}

int main() {
	int n, k;
	cin >> n >> k;
	vector<pair<int, int>> arr(n);
	int s = 0;
	for (int i = 0; i < n; ++i) {
		cin >> arr[i].first >> arr[i].second;
		s += arr[i].first;
		s += arr[i].second;
	}
	int ans = INF;
	for (int mask = 0; mask < (1 << n); ++mask) {
		ans = min(ans, f(arr, k, mask));
	}
	cout << s - ans << endl;
	//system("pause");
	return 0;
}
```