

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ЖАБОК

Нека $A(x_1, y_1)$ и $B(x_2, y_2)$ са две от дадените точки. Най-малкият брой скокове, които трябва да направи Боко за да стигне от едната точка до другата е равен на $\max\{|x_2 - x_1|, |y_2 - y_1|\}$. Това число в следващите разглеждания ще наричаме *разстояние* между двете точки. Броят на дадените точки е най-много 12, поради което могат да се разгледат всички вариации на тези $2n$ точки от n -ти клас. За всяка от вариациите пресмятаме сумата от *разстоянията* от началото на координатната система до първата точка във вариацията и *разстоянията* между всеки две последователни точки във вариацията. Най-малката от тези суми е търсения в задачата най-малък брой скокове.

Реализация на програмата:

```
#include <iostream>
#include <algorithm>
using namespace std;

const int MAXN = 15;
struct Point
{
    int x, y;
};

int n, used[MAXN], b[MAXN];
Point a[MAXN];
int d[MAXN][MAXN], minpath = 20000;

void current_minpath()
{
    int s = 0;
    for (int i = 0; i < n; i++)
        s += d[b[i]][b[i+1]];
    if (minpath > s) minpath = s;
}

void variation(int i)
{
    if (i == n + 1)
    {
        current_minpath();
        return;
    }
    for (int j = 1; j <= 2 * n ; j++)
        if (used[j] == 0)
        {
            b[i] = j;
            used[j] = 1;
            variation(i + 1);
            used[j] = 0;
        }
}
```

```
int max(int x, int y)
{
    if (x > y) return x;
    return y;
}

int main()
{
    cin >> n;
    a[0].x = 0;
    a[0].y = 0;
    for (int i = 1; i <= 2*n; i++)
        cin >> a[i].x >> a[i].y;
    for (int i = 0; i < 2*n; i++)
        for (int j = i+1; j <= 2*n ; j++)
        {
            d[i][j] = max(abs(a[i].x - a[j].x), abs(a[i].y - a[j].y));
            d[j][i] = d[i][j];
        }
    b[0] = 0;
    variation(1);
    cout << minpath << endl;
    return 0;
}
```

Автор: Младен Манев