

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
по «Алгоритмам и структурам данных»

Выполнил:

Студент группы Р3210

Кадыров А.Х.

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2019

Задача №1444 «Накормить слона»

Пояснение к примененному алгоритму:

Так как нет такого-что все точки лежат на одной прямой, то можно сесть все точки. Сместим начало координат в начальную точку. Для каждой точки найдём угол. Отсортируем точки по углам. Найдём две соседние точки - разность углов которых максимальна. И начнём кушать с одной из этих точек. Если начальная точка будет на выпуклой оболочке - то начиная кушать с любой точки - мы сможем наткнуться на свои же следы.

Код:

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

using namespace std;
double PI = 3.141592653;
struct m_point {
    double x, y, angle, dist;
    int id;
    void count_angle_and_dist() {
        if (x > 0) {
            if (y > 0)
                angle = (atan(y / x) * 180.) / PI;
            else if (y == 0)
                angle = 0.;
            else
                angle = ((atan(x / -y) * 180.) / PI) + 270.;
        } else if (x == 0) {
            if (y > 0)
                angle = 90.;
            else
                angle = 270.;
        } else {
            if (y > 0)
                angle = (atan(-x / y) * 180. / PI) + 90.;
            else if (y == 0)
                angle = 180.;
```

```

    else
        angle = (atan(-y / -x) * 180. / PI) + 180.;
    }
    dist = (x * x) + (y * y);
}
} arr[30100];
bool cmp(m_point a, m_point b) {
    if (a.angle < b.angle)
        return 1;
    if ((a.angle == b.angle) && (a.dist < b.dist))
        return 1;
    return 0;
}
int get_start_point(int n) {
    double mx_angle = 360. - arr[n].angle + arr[1].angle;
    int mx_id = 1;
    for (int i = 2; i <= n; i++) {
        if (mx_angle < arr[i].angle - arr[i - 1].angle) {
            mx_id = i;
            mx_angle = arr[i].angle - arr[i - 1].angle;
        }
    }
    return mx_id;
}
int main() {
    int n;
    cin >> n >> arr[0].x >> arr[0].y;
    n--;
    for (int i = 1; i <= n; i++) {
        cin >> arr[i].x >> arr[i].y;
        arr[i].id = i + 1;
        arr[i].x -= arr[0].x;
        arr[i].y -= arr[0].y;
    }
}

```

```
    arr[i].count_angle_and_dist();  
}  
sort(arr + 1, arr + 1 + n, cmp);  
cout << n + 1 << "\n1\n";  
int id = get_start_point(n);  
for (int i = id; i <= n; i++) {  
    cout << arr[i].id << '\n';  
}  
for (int i = 1; i < id; i++) {  
    cout << arr[i].id << '\n';  
}  
}
```