

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

## Алгоритмы и структуры данных

Отчёт по лабораторной работе №8 (1444)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

## 1. Цель работы

Решить задачу №1444 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1444>

## 2. Задача

### Условие

Гарри Поттер сдаёт экзамен по предмету «Уход за магическими существами». Его задание — накормить карликового элфпотамы. Гарри помнит, что элфпотамы отличаются прямолинейностью и невозмутимостью. Они настолько прямолинейны, что ходят строго по прямой, и настолько невозмутимы, что заставить их идти можно, только если привлечь его внимание к чему-нибудь действительно вкусному. И главное, наткнувшись на цепочку своих собственных следов, элфпотам впадает в ступор и отказывается идти куда-либо. По словам Хагрида, элфпотамы обычно возвращаются домой, идя в обратную сторону по своим собственным следам. Поэтому они никогда не пересекают их, иначе могут заблудиться. Увидев свои следы, элфпотам детально вспоминает все свои перемещения от выхода из дома (поэтому-то они и ходят только по прямой и лишний раз не меняют направление — так легче запоминать). По этой информации элфпотам вычисляет, в какой стороне расположена его нора, после чего поворачивается и идет прямо к ней. Эти вычисления занимают у элфпотам некоторое (довольно большое) время. А то, что некоторые невежды принимают за ступор, на самом деле есть проявление выдающихся вычислительных способностей этого чудесного, хотя и медленно соображающего животного!

Любимое лакомство элфпотамов — слоновьи тыквы, именно они и растут на лужайке, где Гарри должен сдавать экзамен. Перед началом испытания Хагрид притащит животное к одной из тыкв. Скормив элфпотаму очередную тыкву, Гарри может направить его в сторону любой оставшейся тыквы. Чтобы сдать экзамен, надо провести элфпотаму по лужайке так, чтобы тот съел как можно больше тыкв до того, как наткнется на свои следы.

Ограничение времени: 0.25 секунды

Ограничение памяти: 64 МБ

### Исходные данные

В первой строке входа находится число  $N$  ( $3 \leq N \leq 30000$ ) — количество тыкв на лужайке. Тыквы пронумерованы от 1 до  $N$ , причем номер один присвоен той тыкве, у которой будет стоять элфпотам в начале экзамена. В следующих  $N$  строках даны координаты всех тыкв по порядку. Все координаты — целые числа от  $-1000$  до  $1000$ . Известно, что положения всех тыкв различны, и не существует прямой, проходящей сразу через все тыквы.

### Результат

В первой строке выхода вы должны вывести  $K$  — максимальное количество тыкв, которое может съесть элфпотам. Далее по одному числу в строке выведите  $K$  чисел — номера тыкв в порядке их обхода. Первым в этой последовательности всегда должно быть число 1.

### Пример

Исходные данные	Результат
4	4
0 0	1
10 10	3
0 10	2
10 0	4

## 3. Материалы работы

### 3.1. Объяснение алгоритма

Для решения задачи первая точка выбирается как начало координат. По ходу считывания записывается угол каждой точки относительно оси  $Ox$  и расстояние до нового начала координат. После этого точки сортируются по углу и по расстоянию до начала координат. После сортировки находится максимальный угол между направлениями на две соседние точки, чтобы найти начало обхода, исключаящее пересечение с маршрутом от 1 до 2 точки. После всех операций выводится обход точек «по кругу».

### 3.2. Код программы.

```
1. #include <iostream>
2. #include <cmath>
3.
4. using namespace std;
5.
6. #define PI 3.14159265;
7.
8. struct point {
9.     float length;
10.    double degrees;
11.    int index;
12.};
13.
14.int compare_points(const void *var1, const void *var2) {
15.    const point*p1 = (point*) var1;
16.    const point*p2 = (point*) var2;
17.    if (p1->degrees - p2->degrees > 1e-10)
18.        return 1;
19.    else if (p1->degrees - p2->degrees < -1e-10)
20.        return -1;
21.    else
22.        if (p1->length > p2->length) return 1;
23.        else return -1;
```

```

24. }
25.
26.
27. int main() {
28.     int n;
29.     short f_x, f_y, x, y;
30.     cin >> n >> f_x >> f_y;
31.     point dots[n];
32.     dots[0].degrees = -1;
33.     dots[0].length = 0;
34.     dots[0].index = 1;
35.     for (int i = 1; i < n; i++) {
36.         cin >> x;
37.         cin >> y;
38.         dots[i].index = i+1;
39.         dots[i].length = (x - f_x)*(x - f_x) + (y - f_y)*(y - f_y);
40.         dots[i].degrees = atan2(y - f_y, x - f_x) * 180.0 / PI;
41.         if (y - f_y < 0) dots[i].degrees += 360;
42.     }
43.
44.     qsort(dots, n, sizeof(point), compare_points);
45.
46.
47.     int start_point = 1;
48.     double max_a = dots[1].degrees - dots[n-1].degrees + 360;
49.     for (int i = 1; i < n - 1; i++) {
50.         if (dots[i + 1].degrees - dots[i].degrees > max_a) {
51.             max_a = dots[i + 1].degrees - dots[i].degrees;
52.             start_point = i+1;
53.         }
54.     }
55.     cout << n << endl << 1 << endl;
56.
57.     for (int i = start_point; i < n; i++) {
58.         cout << dots[i].index << endl;
59.     }
60.     for (int i = 1; i < start_point; i++) {
61.         cout << dots[i].index << endl;
62.     }
63.
64.     return 0;
65. }

```

#### 4. Результат выполнения и ссылка на репозиторий GitHub

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">9326789</a>	14:36:10 18 апр 2021	<a href="#">Daniil Marukhlenko</a>	<a href="#">1444</a>	G++ 9.2 x64	Accepted		0.109	1 124 КБ

[https://github.com/japersik/algorithms\\_and\\_data\\_structures/](https://github.com/japersik/algorithms_and_data_structures/)



#### 5. Вывод

Работа выполнена, задача решена с использованием сортировки структур.