

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №6 (1207)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

1. Цель работы

Решить задачу №1207 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1207>

2. Задача

Условие

На плоскости находятся N точек (N чётно). Никакие три точки не лежат на одной прямой. Ваша задача — выбрать две точки так, что прямая линия, проходящая через них, делит множество точек на две части одинакового размера.

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Исходные данные

Первая строка содержит целое число N ($4 \leq N \leq 10\,000$). Каждая из следующих N строк содержит пары целых чисел x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$) — координаты i -й точки.

Результат

Выведите номера выбранных точек.

Пример

Исходные данные	Результат
4 0 0 1 0 0 1 1 1	1 4

3. Материалы работы

3.1. Объяснение алгоритма

Для решения задачи представим точки на плоскости. После этого найдём самую «левую» точку и переместим в неё начало координат с сохранением направления и размерности осей. После этого необходимо найти тангенсы углов между прямыми, соединяющей начало координат с каждой из точек, и новой осью. После — необходимо отсортировать структуру массив с точками по тангенсу угла и найти «средний» элемент в массиве. В ответ идёт точка, стоящая в начале координат и найденный элемент.

3.2. Код программы.

```
1. #include <iostream>
2. #include <algorithm>
3. #include <limits>
4. using namespace std;
5.
6. struct Point {
7.     int x;
```

```

8.     int y;
9.     int number;
10.    double tang;
11.};
12.
13.bool compare_angle(Point p1, Point p2) { return p2.tang > p1.tang; }
14.
15.int main() {
16.    int n;
17.    cin >> n;
18.    Point points[n];
19.    int minIndex = 0;
20.    for (int i = 0; i < n; i++) {
21.        cin >> points[i].x >> points[i].y;
22.        points[i].number = i + 1;
23.        if (points[i].x < points[minIndex].x)
24.            minIndex = i;
25.    }
26.    int min_x = points[minIndex].x;
27.    int min_y = points[minIndex].y;
28.    for (int i = 0; i < n; i++) {
29.        points[i].x = points[i].x - min_x;
30.        points[i].y = points[i].y - min_y;
31.        if (points[i].x == 0 ) {
32.            if (points[i].y > 0) {points[i].tang = numeric_limits<double>::max();}
33.            else if (points[i].y < 0) {points[i].tang =
numeric_limits<double>::lowest()+0.000000001;}
34.            else if (points[i].y == 0) {points[i].tang =
numeric_limits<double>::lowest();}
35.        }
36.        else points[i].tang = ((double)points[i].y)/((double) points[i].x);
37.    }
38.    }
39.    sort(points, points+n, compare_angle);
40.
41.    cout << minIndex+1<< " " << points[n / 2].number << endl;
42.    return 0;
43.}

```

4. Результат выполнения и ссылка на репозиторий GitHub

9259038	00:20:56 9 мар 2021	Daniil Marukhlenko	1207	G++ 9.2 x64	Accepted		0.046	636 КБ
---------	------------------------	------------------------------------	----------------------	-------------	----------	--	-------	--------

https://github.com/japersik/algorithms_and_data_structures/



5. Вывод

Работа выполнена, задача решена с использованием сортировки структур.