

Отчет

по лабораторной работе «1207. Медиана на плоскости»

по дисциплине «Алгоритмы и Структуры данных»

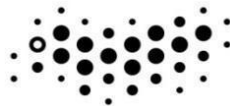
Авторы:

Полит Алексей Денисович

Факультет: СУиР

Группа: R3235

Преподаватель: Тропченко Андрей Александрович



УНИВЕРСИТЕТ ИТМО

1. Задачи

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.

2. 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 ($-106 \leq x_i, y_i \leq 106$) — координаты i -й точки.

3. Текст программы

```
#include <iostream>
#include <cmath>
#define PI 3.141592653589793238462643383279502884197169399375105820974
using namespace std;

struct Point {
    int x;
    int y;
    int number;
    double angle;
};

void swap(Point* a, Point* b) {
    Point t = *a;
    *a = *b;
    *b = t;
}

void quickSort(Point* array, int left, int right) {
    int i = left, j = right;
    double pivot = array[(left + right) / 2].angle;
    while(i <= j) {
        while (array[i].angle < pivot) i++;
        while (array[j].angle > pivot) j--;
        if (i > j) break;
        swap(&array[i], &array[j]);
        i++;
        j--;
    }
    if (left < j) quickSort(array, left, j);
    if (i < right) quickSort(array, i, right);
}

int main() {
    int n;
    cin >> n;
    Point points[n];
```

```

pair<int, int> minPoint = make_pair(INT_MAX, 0);
for (int i = 0; i < n; i++) {
    cin >> points[i].x >> points[i].y;
    points[i].number = i + 1;
    if (points[i].y < minPoint.first || points[i].y == minPoint.first && points[i].x <
points[minPoint.second].x) {
        minPoint.first = points[i].y;
        minPoint.second = i;
    }
}
swap(&points[0], &points[minPoint.second]);
for (int i = 1; i < n; i++) {
    if (points[i].x == points[0].x) points[i].angle = PI / 2;
    else if (points[i].y == points[0].y) points[i].angle = 0;
    else points[i].angle = atan((double)(points[i].y - points[0].y) /
(double)(points[i].x - points[0].x));
    if (points[i].angle < 0) points[i].angle = points[i].angle + 2 * PI;
}
quickSort(points, 1, n - 1);
cout << points[0].number << " " << points[n / 2].number << endl;
return 0;
}

```

9276959 | 13:16:20
22 May 2021 | [Aleksy](#)

[1207. Медиана на плоскости](#)

G++ 9.2 x64

Accepted

0.031

640 КБ

4. Описание программы

Создаём отдельную структуру для хранения сведений о точке (Point). Для того, чтобы прямая линия прошла через две точки и поделила множество точек на две части одинакового размера, для начала нужно выбрать опорную точку (её мы примем за начало координат). Мы возьмём минимальную по ОХ или ОУ точку 1. Построим новые координатные оси. Построим вектора для каждой из введенных точек. Расчитываем углы относительно начала координат. Нужно отсортировать массив точек и выбрать из них среднюю. Прямая, проходящая через опорную и среднюю точки, поделила множество на две части