МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Президентский физико-математический лицей № 239

Отчёт по годовому проекту

Ученик: Зорин Андрей
Преподаватель: Клюнин Алексей Олегович

Класс: 10-3

Санкт-Петербург
2016

Содержание

1 Постановка задачи

Из заданного множества точек на плоскости выбрать 2 точки так, чтобы точки, лежащие по обе сторогы от прямой, соединяющей данные 2 точки, различались наименьшим образом

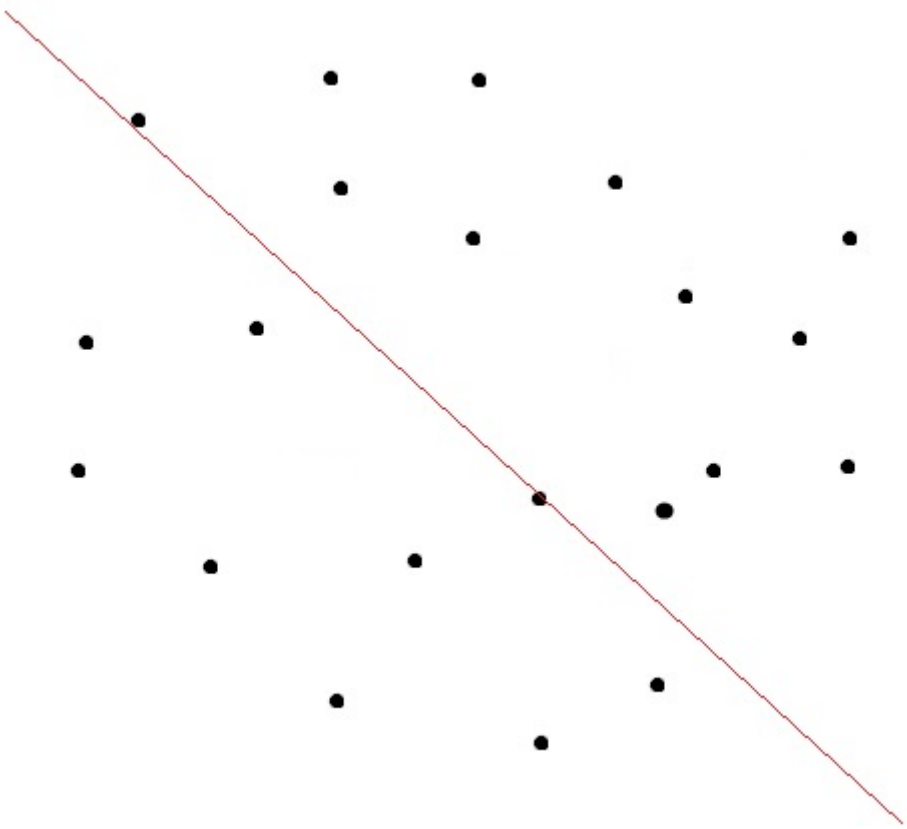

Рис. 1: **Пример подходящей прямой**

2 Алгоритм решения задачи

2.1 Базовые структуры данных

Понадобится 2 класса :
1) Описывает поведение точки.
2) Описывает поведение прямой.

2.1.1 Первый класс (Point) :

Поля : "double x отвечает за координату точки по оси "x";
"double y отвечает за координату точки по оси "y".
1) Сетеры(setX(),setY()).
2) Гетеры(getX(),getY()).
3) Метод "Distance Line(Line ln) возвращает расстояние от точки до данной прямой (ln).
4) Конструкторы :
Point() - создает точку с координатами (0;0);
Point(double x,double y) - создает точку с координатами (x,y).
5) toString - вывод точки в виде : *название точки* (x,y).
Т.к. класс не будет наследоваться, то все методы и поля приватные (private).

2.1.2 Второй класс (Line) :

Поля :
double k - отвечает за тангес угла прямой;
double b - отвечает за
длину отрезка, который отсекает прямая по оси "y считая от начала координат.
1) Сетеры(setX(),setY()).
2) Гетеры(getX(),getY()).
3) Метод "RelationPoint(Point p) возвращает "1 если точка лежит выше прямой, "0 если точка лежит на прямой, 1 если точка лежит ниже прямой.
4) Конструкторы :
Line() - создает прямую "y=x";
Line(double k, double b) - создает прямую "y=kx+b";
Line(Point p1, Point p2) - создает прямую, которой принадлежат две данные точки (p1,p2).
5) "toString вывод прямой в виде : *название прямой* y=k*x+b.
Т.к. класс не будет наследоваться, то все методы и поля приватные (private).

2.2 Построение алгоритма

Рассмотрим всевозможные прямые. Для каждой найдем разницу точек. Найдем минимальную разницу точек среди всех прямых - решим задачу
1) Считываем количество точек (n)
2) Считываем координаты всех точек, создаем объект для каждой из них (с помощью массива объектов).
Всего возможно n*(n-1)/2 разных прямых (если учесть, что на любой прямой лежат только две точки из всех).
3) Рассмотрим каждую прямую, создав цикл "for":
!создавать каждый раз новую прямую не будем. создадим один раз, дальше будем только изменять ее коэффициенты!
Определим взаимное расположение точки, не принадлежащей рассматриваемой прямой, и этой прямой. Заметим, что если сложить все значения этих точек "RelationsPoint"и взять модуль, то получим разницу количества точек, лежащих ниже прямой, и точек, лежащих выше прямой. Соответственно, задача сводится к поиску минимума (для этого создадим переменную min) модуля суммы "RelationPoint"каждой точки к этой прямой (а для этого создадим переменную abs).
Для первой рассматриваемой прямой минимум и есть "abs для остальных же, если min>abs, то min присваиваем abs, переменной "a"присвоем "getK()"прямой, переменной "b"присвоем "getY()"прямой (Нужно запомнить коэффициенты прямой, т.к на данный момент она искомая).
4)Обнуление всех счетчиков, например abs, конец цикла.
5)Вывод искомой прямой.
6)Конец программы.