ГБОУ "Президентский ФМЛ № 239"

Проверка на то, что каждая точка множества лежит хотя бы в двух

правильных треугольниках.

Проект подготовил

Ученик 10-1 класса

Добрис Даниил

1.Постановка задачи.

Множество точек на плоскости назовем дваждытреугольным, если каждая точка этого множества является вершиной хотя бы двух правильных треугольников, построенных по точкам множества. Определить, удовлетворяет ли заданное множество точек этому свойству.

2.Входные и выходные данные.

2.1.Входные данные.

В файле points.txt вводится координаты точек на плоскости. Координаты лежат в промежутке от -1 до 1.

2.2.Выходные данные

На консоль выводится верно ли, что наше множество удовлетворят условию дваждытреугольности.

3.Математическая модель.

Единственная математическая формула, которая нам понадобится - формула для подсчета расстояния между точками A(x1,y1) и B(x2,y2).

4.Анализ структуры данных.

Точки у нас хранятся в массиве ArrayPoints, а треугольники мы ни где не храним так как при нахождении какого-то треугольника, мы просто увеличиваем параметр и потом в конце просто считаем и убираем всё лишнее.

3.Выбор метода решения.

Для начала мы делаем конструктор, который позволит нам рисовать треугольники(в условии задачи конечно не сказано, что нужно выводить треугольники, но на простых случаях это очень помогает понять какие именно равносторонние треугольники получились).После этого мы делаем конструктор, который позволяет нам проверить треугольник на правильность. Задаём параметр a, который будет считать количество равносторонних треугольников для каждой точки. Мы рассматриваем каждую точку и для неё рассматриваем всевозможные равносторонние треугольники, если какой-то треугольник уже был встречен, то его мы не учитываем. Если для какой-то точки параметр меньше двух, то сразу будет выведен false и дальнейшие точки уже бесполезно учитывать, а если мы прошли по всем точкам и для них всех параметр хотя бы 2,то выводим true.

5.Листинг программы.

5.1.Класс отвечающий за точку.

public class Vector2 {

public double x;//задаём поле координаты по оси X

public double y;//задаём поле координаты по оси Y

public Vector2(double x,double y){

this.x=x;

this.y=y;

}

//задаём класс который опишет положение нашей точки на плоскости

public Vector2(Vector2 p){

this.x=p.x;

this.y=p.y;

}

//если нам известна точка, то не ищем её координаты, а сразу записываем в класс

}

5.2.Класс отвечающий за треугольник.

public class Triangle {

public Vector2 a;//поле для первой точки

public Vector2 b;//поле для второй тчоки

public Vector2 c;//поле для третьей точки

public Triangle(Vector2 a, Vector2 b, Vector2 c) {

this.a = a;

this.b = b;

this.c = c;

}

//задаём треугольник с вершинами в точках a,b,c

public Triangle(Point a, Point b, Point c) {

this.a = new Vector2(a.x, a.y);

this.b = new Vector2(b.x, b.y);

this.c = new Vector2(c.x, c.y);

}

//подставляем какие-то три точки из нашего списка,тем самым задавая его точное положеие в пространстве

public void render(GL2 gl) {

Figure.renderTriangle(gl, a, b, c, false);

}

//рисуем наш треугольник

public boolean regular() {

double r1 = Math.sqrt((a.x - b.x) \* (a.x - b.x) + (a.y - b.y) \* (a.y - b.y));//расстояние между первыми двумя точками

double r2 = Math.sqrt((c.x - b.x) \* (c.x - b.x) + (c.y - b.y) \* (c.y - b.y));//расстояние между вторыми двумя точками

double r3 = Math.sqrt((a.x - c.x) \* (a.x - c.x) + (a.y - c.y) \* (a.y - c.y));//расстояние между оставшимися двумя точками

return Math.abs(r1 - r2) < 0.01 && Math.abs(r1 - r3) < 0.01 && Math.abs(r2 - r3) < 0.01;//проверяем все 3 расстояния на равенство между собой

}

}

5.3.Класс рисования.

public class Figure {//класс, отвечающий за рисование

public static void renderPoint(GL2 gl,Vector2 pos,double size){//рисуем точку

gl.glPointSize((float) size);//задаём размер точки

gl.glBegin(GL2.GL\_POINTS);//оператор рисования точки

gl.glVertex2d(pos.x,pos.y);//задаём положение точкм

gl.glEnd();//конец рисования

}

public static void renderLine(GL2 gl,Vector2 posA,Vector2 posB,double width){//рисуем отрезок

gl.glLineWidth((float) width);//ширина линии отрезка

gl.glBegin(GL2.GL\_LINES);//оператор рисования отрезка

gl.glVertex2d(posA.x,posA.y);//начало отрезка

gl.glVertex2d(posB.x,posB.y);//конец отрезка

gl.glEnd();//конец рисования

}

public static void renderTriangle(GL2 gl,Vector2 posA,Vector2 posB,Vector2 posC,boolean filled) {//рисуем треугольник

if(!filled){//закрашенность треугольника

gl.glBegin(GL2.GL\_LINE\_STRIP);//оператор рисования линий по указанному порядку точек

gl.glVertex2d(posA.x,posA.y);//1 точка следования

gl.glVertex2d(posB.x,posB.y);//2 точка следования

gl.glVertex2d(posC.x,posC.y);//3 точка следования

gl.glVertex2d(posA.x,posA.y);//1 точка следования,так как нам нужно вернуться в начало,чтобы замкнуть треугольник

gl.glEnd();//конец рисования

}

}

}

5.4.Решение задачи

public void solve() {//решение задачи

triangles.clear();//очищаем картинку от треугольников в самом начале программы

boolean flg = true;

int a = 0;//вводим параметр,который будет считать треугольники для каждой точки

//в следующих строках для каждой точки считаем количество правильных треугольниках

for (int i = 0; i < points.size(); i++) {

Point p = points.get(i);//номер 1 точки

for (int j = 0; j < points.size(); j++) {

Point p2 = points.get(j);//номер второй точки

for (int k = 0; k < points.size(); k++) {

if (i == j || j == k || k == i)//проверка на то,что такой треугольник уже был

continue;//если был такой треугольник,то записываем его снова

Point p3 = points.get(k);//номер третьей точки

Triangle t = new Triangle(p, p2, p3);//создаём новый треугольник из этих трёх точек

if (t.regular()) {//проверка на правильность треугольника

triangles.add(t);//если он правильный,то добавляем его на картинку

a++;//увеличиваем число правильных треугольников для 1 точки

}

}

}//так как какие-то треугольники мы считали дважды, то мы будём не a, а a/2

if (a / 2 < 2) {//проверка на то,что условие дваждытреугольности выполняется

flg = false ;

}

a = 0;//чтобы рассматриваь другую точку заного,не учитывая предыдущих, нужно обнулить параметр

}

System.out.println(flg);//узнаём,является ли данное множество дваждытреугольным

}

6.Пример работы программы

Входные данные Выходные данные

0,35 0,59 true

-0,43 0,59

0,36 -0,76

-0,42 -0,76

-0,81 -0,09

-0,04 -0,08