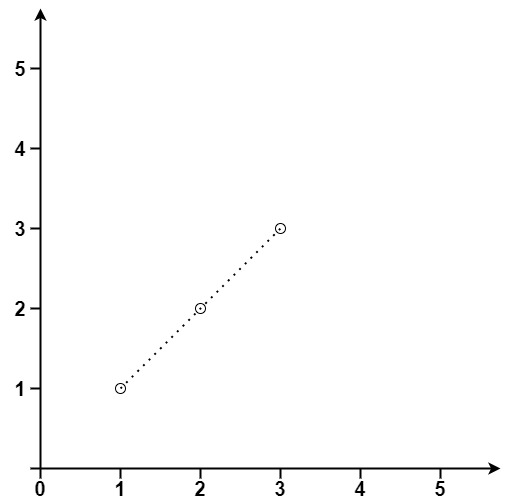
Given an array of points where points[i] = [xi, yi] represents a point on the **X-Y** plane, return *the maximum number of points that lie on the same straight line*.

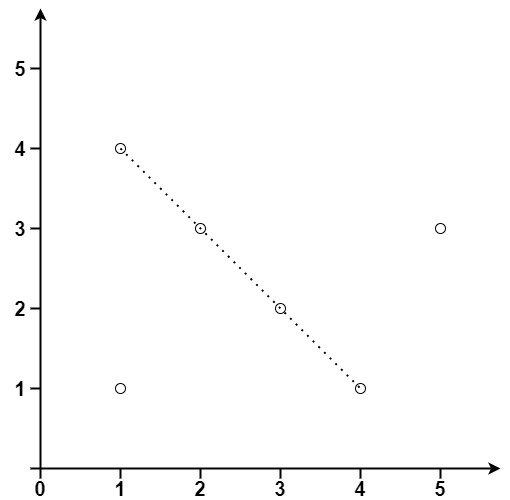
**Example 1:**



**Input:** points = [[1,1],[2,2],[3,3]]

**Output:** 3

**Example 2:**



**Input:** points = [[1,1],[3,2],[5,3],[4,1],[2,3],[1,4]]

**Output:** 4

Solution:

class Solution {

public int maxPoints(int[][] points) {

if (points==null) return 0;

if (points.length<=2) return points.length;

Map<Integer,Map<Integer,Integer>> map = new HashMap<Integer,Map<Integer,Integer>>();

int result=0;

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

map.clear();

int overlap=0,max=0;

for (int j=i+1;j<points.length;j++){

int x=points[j][0]-points[i][0];

int y=points[j][1]-points[i][1];

if (x==0&&y==0){

overlap++;

continue;

}

int gcd=generateGCD(x,y);

if (gcd!=0){

x/=gcd;

y/=gcd;

}

if (map.containsKey(x)){

if (map.get(x).containsKey(y)){

map.get(x).put(y, map.get(x).get(y)+1);

}else{

map.get(x).put(y, 1);

}

}else{

Map<Integer,Integer> m = new HashMap<Integer,Integer>();

m.put(y, 1);

map.put(x, m);

}

max=Math.max(max, map.get(x).get(y));

}

result=Math.max(result, max+overlap+1);

}

return result;

}

private int generateGCD(int a,int b){

if (b==0) return a;

else return generateGCD(b,a%b);

}

}