

0149. 直线上最多的点数

👤 ITCharge ⌚ 大约 2 分钟

- 标签：几何、数组、哈希表、数学
- 难度：困难

题目链接

- [0149. 直线上最多的点数 - 力扣](#)

题目大意

给定一个平面上的 n 个点的坐标数组 `points`，求解最多有多少个点在同一条直线上。

解题思路

两个点可以确定一条直线，固定其中一点，求其他点与该点的斜率，斜率相同的点则在同一条直线上。可以考虑把斜率当做哈希表的键值，存储经过该点，不同斜率的直线上经过的点的数目。

对于点 i ，查找经过该点的直线只需要考虑 $(i+1, n-1)$ 位置上的点即可，因为 $i-1$ 之前的点已经在遍历点 $i-2$ 的时候考虑过了。

斜率的计算公式为 $\frac{dy}{dx} = \frac{y_j - y_i}{x_j - x_i}$ 。

因为斜率是小数会有精度误差，所以我们考虑使用 (dx, dy) 的元组作为哈希表的 `key`。

注意：

需要处理倍数关系， dy 、 dx 异号情况，以及处理垂直直线（两点横坐标差为 0）的水平直线（两点横坐标差为 0）的情况。

代码

py

```
class Solution:
    def maxPoints(self, points: List[List[int]]) -> int:
        n = len(points)
        if n < 3:
            return n
        ans = 0
        for i in range(n):
            line_dict = dict()
            line_dict[0] = 0
            same = 1
            for j in range(i+1, n):
                dx = points[j][0] - points[i][0]
                dy = points[j][1] - points[i][1]
                if dx == 0 and dy == 0:
                    same += 1
                    continue
                gcd_dx_dy = math.gcd(abs(dx), abs(dy))
                if (dx > 0 and dy > 0) or (dx < 0 and dy < 0):
                    dx = abs(dx) // gcd_dx_dy
                    dy = abs(dy) // gcd_dx_dy
                elif dx < 0 and dy > 0:
                    dx = -dx // gcd_dx_dy
                    dy = -dy // gcd_dx_dy
                elif dx > 0 and dy < 0:
                    dx = dx // gcd_dx_dy
                    dy = dy // gcd_dx_dy
                elif dx == 0 and dy != 0:
                    dy = 1
                elif dx != 0 and dy == 0:
                    dx = 1
                key = (dx, dy)
                if key in line_dict:
```

```
        line_dict[key] += 1
    else:
        line_dict[key] = 1
    ans = max(ans, same + max(line_dict.values()))
return ans
```