149. Max Points On a Line

Given n points on a 2D plane, find the maximum number of points that lie on the same straight line.

找到给定平面中在一条直线上的最多点数。

C++:

```
/**
 * Definition for a point.
 * struct Point {
 *         int x;
 *         int y;
 *         Point() : x(0), y(0) {}
 *         Point(int a, int b) : x(a), y(b) {}
 * };
 */

int maxPoints(vector<Point>& points) {
        int len = points.size(), ans = 0;
        if(len <= 2) return len;
        unordered_map<double, int> mp;
        for(int i = 0; i < len; i++){
            int dup = 1;        // To count the points exactly same with point i, including i itself.
            mp.clear();
            for(int j = i+1; j < len; j++){     // we start with j = i +1, because lines from i to k (k = 0,1,...,i-1) has already been considered in the previous steps.
                if(points[i].x == points[j].x && points[i].y == points[j].y){
                    dup++;
                    continue;
                }
                else{
                    float slope = (points[i].x == points[j].x) ? INT_MAX : (float(points[i].y-points[j].y)/(points[i].x-points[j].x));
                    mp[slope]++;
                }
            }
            ans = max(dup , ans);    // in case no element in the map.
            for (auto slope : mp)
                if (slope.second + dup > ans)
```

```
                    ans = slope.second + dup;
                }
                  return ans;
}
```

JAVA:

```
 /**
 * Definition for a point.
 * class Point {
 *         int x;
 *         int y;
 *         Point() { x = 0; y = 0; }
 *         Point(int a, int b) { x = a; y = b; }
 * }
 */
public class Solution {
      public int maxPoints(Point[] points) {
            if(points.length <= 0) return 0;
            if(points.length <= 2) return points.length;
            int result = 0;
            for(int i = 0; i < points.length; i++){
                  HashMap<Double, Integer> hm = new HashMap<Double, Integer>();
                  int samex = 1;
                  int samep = 0;
                  for(int j = 0; j < points.length; j++){
                        if(j != i){
                              if((points[j].x == points[i].x) && (points[j].y == points[i].y)){
                                    samep++;
                              }
                              if(points[j].x == points[i].x){
                                    samex++;
                                    continue;
                              }
                              double k = (double)(points[j].y - points[i].y) /
(double)(points[j].x - points[i].x);
                              if(hm.containsKey(k)){
                                    hm.put(k,hm.get(k) + 1);
                              }else{
                                    hm.put(k, 2);
                              }
                              result = Math.max(result, hm.get(k) + samep);
                        }
```

```
                }
                result = Math.max(result, samex);
            }
            return result;
        }
}
```

PYTHON:

```
# Definition for a point.
# class Point(object):
#         def __init__(self, a=0, b=0):
#             self.x = a
#             self.y = b

def maxPoints(self, points):

  """
            :type points: List[Point]
            :rtype: int
  """
        answer = 0
        for p in points:
            pctr = 0
            ctr = collections.Counter()
            for q in points:
                x, y = q.x - p.x, q.y - p.y
                pctr += x == y == 0
                ctr[float(y)/x if x else 'inf'] += 1
            ctr['inf'] -= pctr
            answer = max(answer, pctr + max(ctr.values()))
        return answer
```