Approach 1: Brute force solution

1) for each point, compare it with every other point and find all slopes that can be generated.

slope = (y2-y1) / (x2-x1) except (x2-x1)=0 then slope is inf

create a 2d-hashMap slope which stores all slopes that can be generated for a point when compared with other points.

slopes[(x,y)]

if slope exists increment frequency of slope, update max\_pts

2) return max\_pts

Time -> O(n^2)

Space -> O(n) (store slope values)

Code:

class Solution:

def maxPoints(self, points: List[List[int]]) -> int:

if len(points) < 3:

return len(points)

slope = {}

n = len(points)

max\_pts = 1

for i in range(n):

x1,y1 = points[i]

slope[(x1,y1)] = {}

for j in range(i+1, n):

x2,y2 = points[j]

dx = x2-x1

dy = y2-y1

# vertical aligned

if dy == 0:

m = 0

# horizontally aligned

elif dx == 0:

m = '-inf'

else:

m = dy/dx

# map points to slope

if m in slope[(x1,y1)]:

slope[(x1,y1)][m] +=1

max\_pts = max(max\_pts, slope[(x1,y1)][m])

else:

slope[(x1,y1)][m] = 1

return max\_pts + 1