

Count maximum points on same line

Given N point on a 2D plane as pair of (x, y) co-ordinates, we need to find maximum number of point which lie on the same line.

Examples:

```
Input : points[] = {-1, 1}, {0, 0}, {1, 1},  
                  {2, 2}, {3, 3}, {3, 4}  
  
Output : 4  
Then maximum number of point which lie on same  
line are 4, those point are {0, 0}, {1, 1}, {2, 2},  
{3, 3}
```

Recommended: Please solve it on “PRACTICE” first, before moving on to the solution.

We can solve above problem by following approach – For each point p, calculate its slope with other points and use a map to record how many points have same slope, by which we can find out how many points are on same line with p as their one point. For each point keep doing the same thing and update the maximum number of point count found so far.

Some things to note in implementation are:

1) if two point are (x1, y1) and (x2, y2) then their slope will be $(y2 - y1) / (x2 - x1)$ which can be a double value and can cause precision problems. To get rid of the precision problems, we treat slope as pair $((y2 - y1), (x2 - x1))$ instead of ratio and reduce pair by their gcd before inserting into map. In below code points which are vertical or repeated are treated separately.

2) If we use `unordered_map` in c++ or `HashMap` in Java for storing the slope pair, then total time complexity of solution will be $O(n^2)$

[Learn More](#)
Real-time messaging, file sharing and powerful search. Slack: where work happens.

```
/* C/C++ program to find maximum number of point
   which lie on same line */
#include <bits/stdc++.h>
using namespace std;

// method to find maximum colinear point
int maxPointOnSameLine(vector< pair<int, int> > points)
{
    int N = points.size();
    if (N < 2)
        return N;

    int maxPoint = 0;
    int curMax, overlapPoints, verticalPoints;

    // map to store slope pair
    unordered_map<pair<int, int>, int> slopeMap;

    // looping for each point
    for (int i = 0; i < N; i++)
    {
        curMax = overlapPoints = verticalPoints = 0;

        // looping from i + 1 to ignore same pair again
        for (int j = i + 1; j < N; j++)
        {
            // If both point are equal then just
            // increase overlapPoint count
            if (points[i] == points[j])
                overlapPoints++;

            // If x co-ordinate is same, then both
            // point are vertical to each other
            else if (points[i].first == points[j].first)
                verticalPoints++;

            else
            {
                int yDif = points[j].second - points[i].second;
                int xDif = points[j].first - points[i].first;
                int g = __gcd(xDif, yDif);

                // reducing the difference by their gcd
                yDif /= g;
                xDif /= g;

                // increasing the frequency of current slope
                // in map
                slopeMap[make_pair(yDif, xDif)]++;
                curMax = max(curMax, slopeMap[make_pair(yDif, xDif)]);
            }

            curMax = max(curMax, verticalPoints);
        }

        // updating global maximum by current point's maximum
        maxPoint = max(maxPoint, curMax + overlapPoints + 1);

        // printf("maximum colinear point which contains current
```

```

        // point are : %d\n", curMax + overlapPoints + 1);
        slopeMap.clear();
    }

    return maxPoint;
}

// Driver code
int main()
{
    const int N = 6;
    int arr[N][2] = {{-1, 1}, {0, 0}, {1, 1}, {2, 2},
                    {3, 3}, {3, 4}};

    vector< pair<int, int> > points;
    for (int i = 0; i < N; i++)
        points.push_back(make_pair(arr[i][0], arr[i][1]));

    cout << maxPointOnSameLine(points) << endl;

    return 0;
}

```



Learn More

Real time messaging, file sharing and powerful search. Slack: where work happens.

Run on IDE

Output:

4

This article is contributed by **Utkarsh Trivedi**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Learn how to **grow**
your business online.

REGISTER NOW

Google





**EXPERTISE TO HELP
OPTIMIZE YOUR AWS COST
AND PERFORMANCE.**

Real time messaging, file sharing and powerful search. Slack: where work happens.

GET A FREE QUOTE

rackspace.

aws CERTIFIED
AWS PARTNER NETWORK



slack

Learn More

Real time messaging, file sharing and powerful search. Slack: where work happens.

GATE CS Corner Company Wise Coding Practice

Geometric Hash

Recommended Posts:

Group Shifted String

Count Substrings with equal number of 0s, 1s and 2s

Line Clipping | Set 1 (Cohen–Sutherland Algorithm)

Non-crossing lines to connect points in a circle

Maximum number of squares that can fit in a right angle isosceles triangle

Minimum Perimeter of n blocks

Check if a line touches or intersects a circle

Find if it's possible to rotate the page by an angle or not.

Check whether given circle resides in boundary maintained by two other circles

Check if two given circles touch or intersect each other

(Login to Rate and Mark)

4.2

Average Difficulty : **4.2/5.0**
Based on **26** vote(s)

Add to TODO List

Mark as DONE



Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments



Share this post!

X

@geeksforgeeks, [Some rights reserved](#) [Learn More](#) [Contact Us!](#) [About Us!](#) [Careers!](#)

Real time message, file sharing and powerful search. Slack, where work happens.

[Privacy Policy](#)

