Locating vertically and horizontally aligned points in XY Cartesian space.

Ritu Kiran Murmu
IIB2019025

Atithi Kumari
IIB2019026

Shahid
IIB2019027

Abstract: In this paper, we have designed an algorithm to locate vertically aligned points and horizontally aligned points with XY coordinates in a Cartesian space. The algorithm will result in all the starting and ending points of lines which are formed vertically and horizontally.

I. Introduction

We have been given N points in the Cartesian space. In order to find the horizontal and vertical aligned points polynomials, we need to locate all the vertical and horizontal lines in the cartesian space.

We know that to make a line, we need at least 3 points in XY Cartesian space. So, by using this condition, we are going to find all the horizontal and vertical lines in the Cartesian space. The vertical lines can be found by grouping those points which have same x-coordinates, and horizontal lines by same y-coordinates.

Map Maps are associative containers that store elements formed by a combination of a key value and a mapped value, following a specific order.

In a map, the key values are generally used to sort and uniquely identify the elements, while the mapped values store the content associated to this key. The types of key and mapped value may differ, and are grouped together.

The time complexity to store values in map is O(Log n) while for unordered map, it is O(1) on average.

The algorithm is described in detail in **part II**.

II. Algorithm Description and Analysis

Solution is based on the hash map as we have to locate the horizontal and vertical align line.

We have used simple logic for vertical align point general form x = K. The x axis has the equation y = 0 taken as reference. And same for horizontal align pont general form y=k.

Data Structures Used:

- Map of vector to store Points
- Map of vector to store the set of Points

1) Generated the random integer value greater than 50 for no. of point.

2) Run the for loop no. of points time and generated the x , y coordinate for the point.

3) In the hash map for the vertical align point store the y coordinate value corresponding to the given x key.

4) In the hashmap for the horizontal align point store the x coordinate value corresponding to the given y key.

5) Traverse the map for x key if more than or equal to 3 y coordinates are available print the start and end point of the vertical align point.

6) Traverse the map for y key if more than or equal to 3 x coordinates are available print the start and end point of the horizontal align point.

For example Let there be 10 points,
(0 , 4) , (4 , 8) , (6 , -6) (-2,-8), (3 , 4) , (3 , -6) , (-1 , 4) , (-13 , -6) , (-8 , -6), (3 , 5)

The horizontal lines formed by given points are: -
(0 , 4),(3 , 4),(-1 , 4)
(6 , -6),(3 , -6),(-8 , -6),(-13 , -6)

The vertical lines formed by given points are: -
(3 , 4),(3 , -6),(3 , 5)

As the start and points of lines are our output, so the output for the set of points will be: -

(0 , 4) (-1 , 4)
(6 , -6) (-13 , -6)

(3 , 4) (3 , 5)

III. Pseudo Code

```
Int:
Function main()
    map<int, vector<int> > mx;
    mx[x].push_back(y);

    for( const auto& pair : mx )
        {
            int s=0;
            int min =INT_MAX;
            int max= INT_MIN;
            for( double d : pair.second )

          {
            if(d > max)
               max=d;
            if(d<min)
               min=d;
            s++;
          }

          if(s>=3)
            {
              print (pair.first "," min"␣"
                     pair.first "," max)

              print("\n")
            }

        }
```

IV. Time Complexity

The time complexity will be equal to O(nlogn).
T(n) = O(n) * log(n) +c

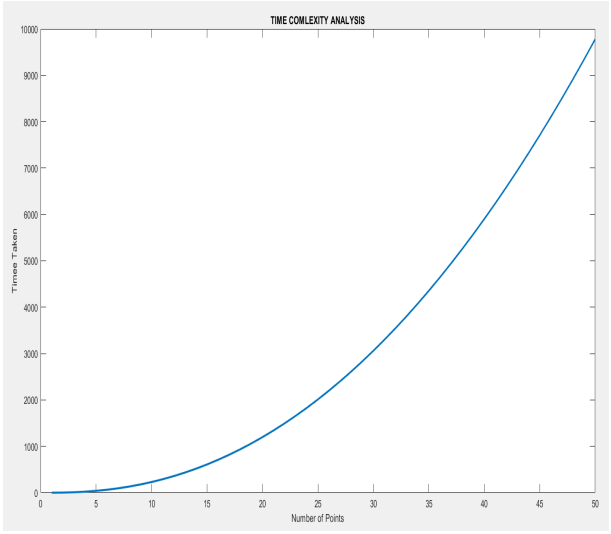

Fig. 1. Time complexity curve

V. Auxiliary Space Complexity

The space complexity of the program is O(n). This is because we are using vectors to store the points.

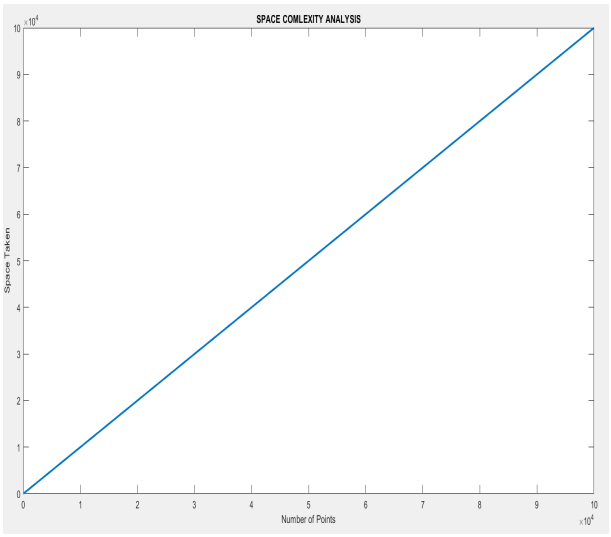

Fig. 2. Space complexity curve

VI. Conclusion

To find the horizontal and vertical align point we have used the hashmap to store the y or x value to x or y key. Time complexity for this program is nlogn and space complexity is n.

VII. References

1) Map of vector

2) Horizontal and vertical lines