Todays Content

1. No:f unique pairs
2. No:f right triangle
3. No:f rectangle { If Time permits }

2Q Given N, 2d points calculate no: of distinct points?

Inputs: 2 arrays x[N] & y[N] given where i^{th} point = x[i] and y[i]

Ex1:

	0	1	2	3	4	5
x[6] = {	2	1	3	2	3	2 }
y[6] = {	3	1	2	5	2	3 }

#Distinct = 4

i^{th} point = (x[i], y[i]);

Idea1: Insert each point in hashmap

#Way1: (x, y) : hm[x] = y

points: (2,3) (1,1) (2,5) (2,3)

hm[2] = 3

hm[1] = 1

hm[2] = 5 ; <2,3> → <2,5>

hm[2] = 3 ; <2,5> → <2,3>

HM

[<2,~~3~~> <1,1>]

Issue: Will not work, will just override existing value

#Way2: Make each point as pair & insert in hashset.

Hashset< pair<int, int> > hs;

↳ Only primitive type & pointers are possible

Possible: Override hashfunction, Next Semester

#Way3: Combine point (x,y) into string & Insert in hashset<String>

P1 = (23, 5) = #Comb1 = 235 #We cannot differentiate

P2 = (2, 35) = #Comb2 = 235 both P1 & P2.

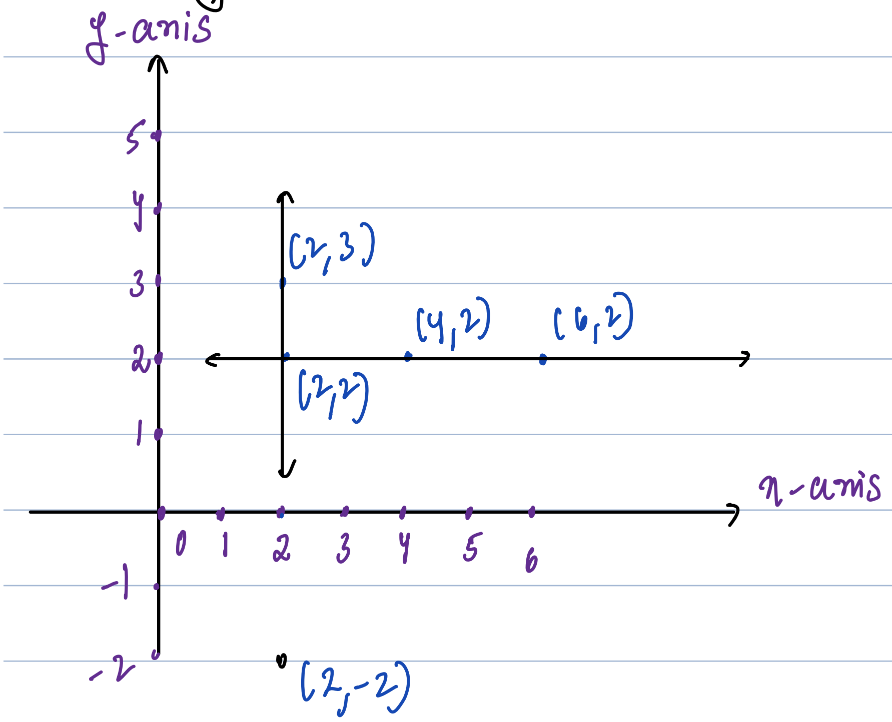
#Note: When we combine P1 & P2 keep a separator.

P1 = (23, 5) = #Com1 = 23@5

P2 = (2, 35) = #Comb2 = 2@35

Return hs.size(), It will give no: of distinct points

// Geometry Basics



Q

1. line parallel to x-axis : x changes y fixed
2. line parallel to y-axis : x fixed y change

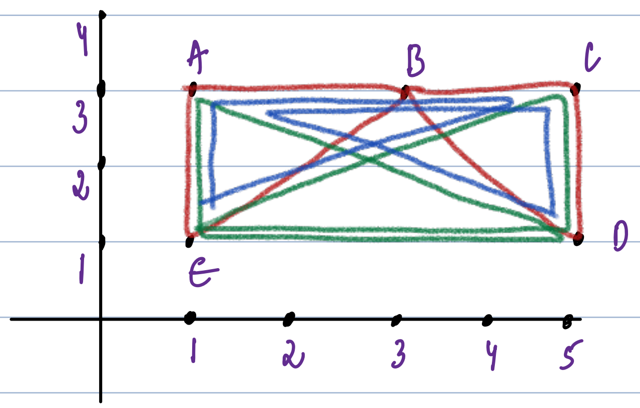
Given N distinct points, calculate how many Right angle Triangles are formed such that shorter sides are parallel to x-axis & y-axis

Inputs: 2 arrays x[N] & y[N] given where i^{th} point : x[i] and y[i]

Points:

$$\begin{array}{llllll} & 0 & 1 & 2 & 3 & 4 \\ x[5] = \{ & 1 & 3 & 5 & 5 & 1 \} \\ y[5] = \{ & 3 & 3 & 3 & 1 & 1 \} \end{array}$$

Ex:



Right Angle Triangle

EAB EAC

ACD BCD

AED CED

Ideas:

Fix 3 points

Generate all triplets:

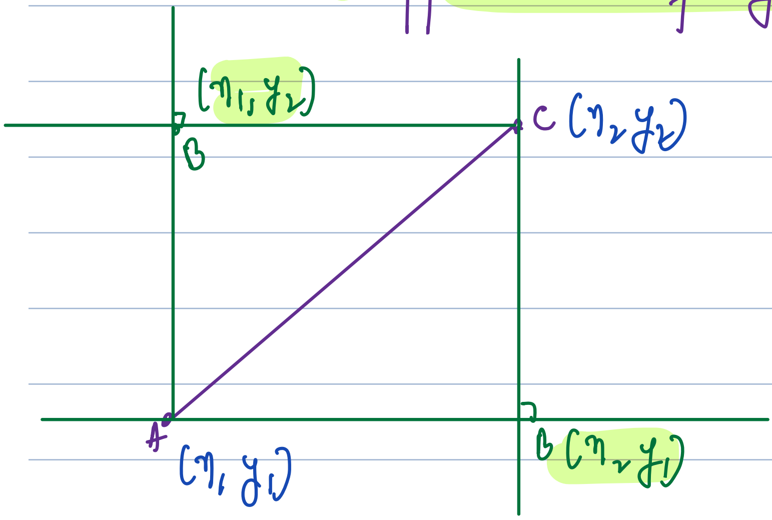
For all triplets, check if we can form right angled triangle parallel to x-axis & y-axis

TC: O(N3) * O(1) SC: O(1)

↳ To check if it forms right angled triangle parallel to x-axis & y-axis

Hint2: Fin 2 points

#Case1: Fin 2 opposite sides of hypotheny

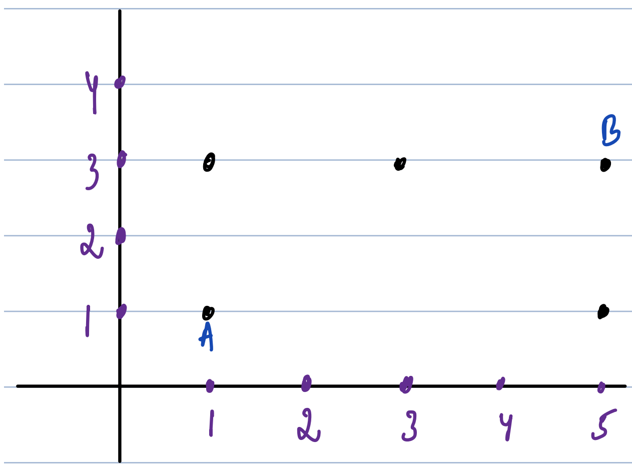


#obs1: When we fin 2 opposite side of hypotheny say (x_1, y_1) & (x_2, y_2)
if (x_1, y_2) exist c+=1
if (x_2, y_1) exist c+=1

#obs2: Hypotheny cannot be parallel to x-axis & y-axis.

#DryRun:

0	1	2	3	4
(1,3)	(3 3)	(5,3)	(1 1)	(5 1)



#Fin 2 points

$x_1\ y_1$	$x_2\ y_2$	$x_2\ y_1$	$x_1\ y_2$
(1,3)	(3,3)	* parallel to x	
(1,3)	(5,3)	* parallel to x	
(1,3)	(1,1)	* parallel to y	
(1,3)	(5,1)	(5,3)	(1,1)
(3,3)	(5,3)	* parallel to x	
(3,3)	(1,1)	(1,3)	(3,1)
(3,3)	(5,1)	(5,3)	(3,1)
(5,3)	(1,1)	(1,3)	(5,1)
(5,3)	(5,1)	* parallel to y	
(1,1)	(5,1)	* parallel to x	

#Note: Store all points has a string in a hashset, so that searching for a point in hashset becomes O(1)

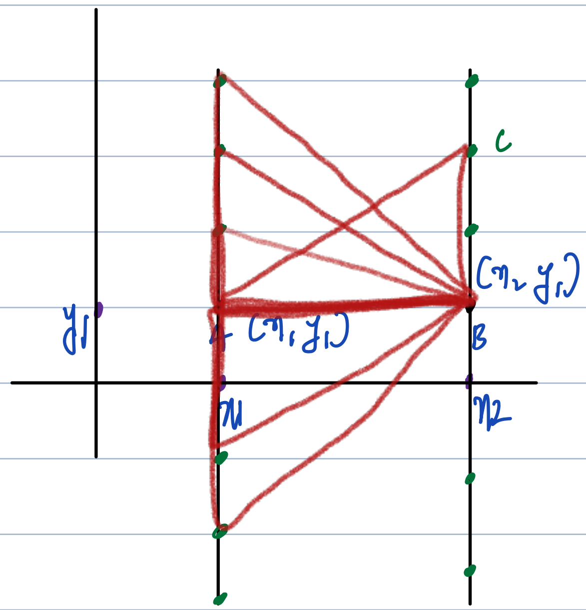
TC: $O(N^2)$

↳ # All pairs, make sure 2 points not parallel to x & y axis.

SC: $O(N)$

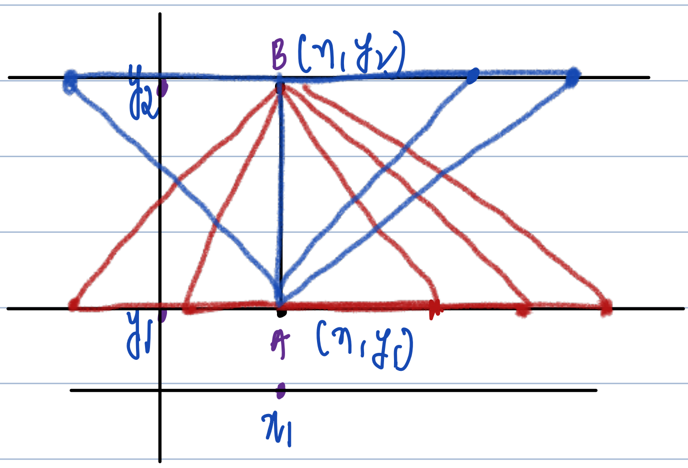
↳ # Store all points in hashset

Case2:



#obs2: If we fix 2 points parallel to x-axis, it's complicated to fix 3rd point

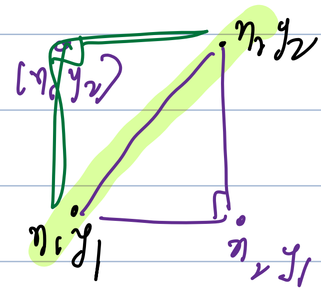
Case3:



#obs3: If we fix 2 points parallel to y-axis, it's complicated to fix 3rd point

```
int Triangu(int x[], int y[]){ TC
    unordered_set<String> hs;
    for(int i=0; i<N; i++){
        hs.insert(x[i]+" "+y[i]);
    }

    int c=0;
    for(int i=0; i<N; i++){
        for(int j=i+1; j<N; j++){
            x1 = x[i], y1 = y[i], x2 = x[j], y2 = y[j];
            # (x1 y1) & (x2 y2)
            if(x1 != x2 && y1 != y2){
                #3 point (x1 y2) (x2 y1)
                if(hs.search(x2+" "+y1)){
                    c++;
                }
                if(hs.search(x1+" "+y2)){
                    c++;
                }
            }
        }
    }
    return c;
}
```



Hint3 : Fin 1 point, which is right angled.

2 smaller sides should be parallel to x-axis & y-axis.

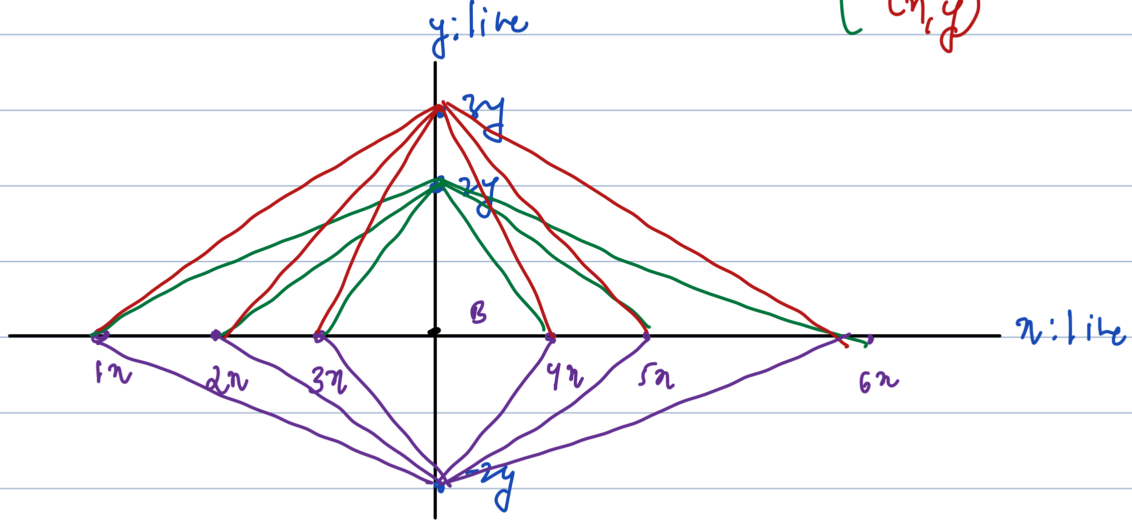
#Obs: Any point along x line & Any point along y line will form ◺

#Con: Calculate no: of points along x line : C_x

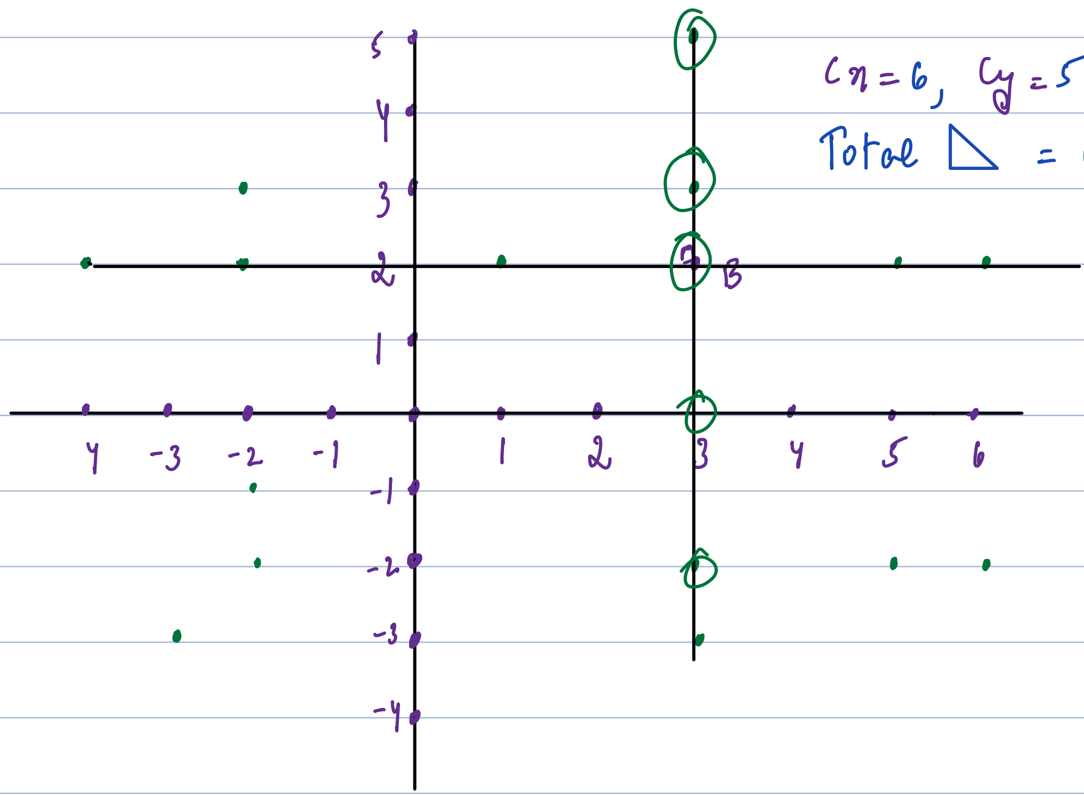
Calculate no: of points along y line : C_y

Total ◺ $= (C_x - 1)(C_y - 1)$

−1, because we neglect (x, y)



Q Calculate how many right angled triangles are there with $(3,2)$ as $90°$ vertex & shorter sides are parallel to x-axis & y-axis



$C_x = 6, \; C_y = 5$

Total ◺ $= (C_x - 1) * (C_y - 1) = 20$

Idea3:

Create 2 hashmaps hmx & hmy.

Insert all points x in hmx

Insert all points y in hmy

Iterate on all points;

\# Consider each point as 90° vertex.

Calculate no: of points along x line = Cx

Calculate no: of points along y line = Cy

ans = ans + (Cx-1) * (Cy-1)

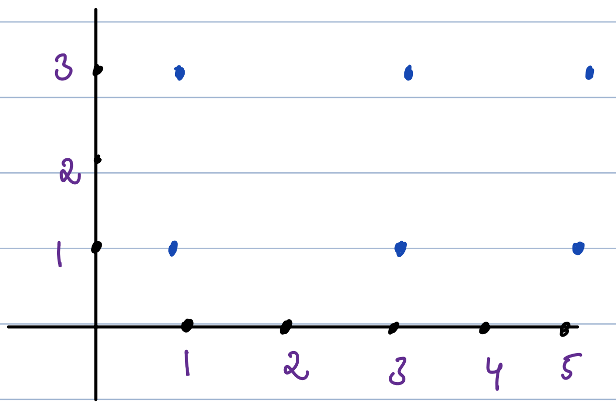
```
int Triangle(vector<int> &x, vector<int> &y){  TC:O(N) SC:O(N)
    Unordered_map <int, int> hx;
    Unordered_map <int, int> hy;
    for(int i=0; i<N; i++){
        #(x[i], y[i])
        hx[x[i]]++;
        hy[y[i]]++;
    }
    int c=0;
    for(int i=0; i<N; i++){
        # (x[i], y[i]) is 90° angle
        int Cx = hx[x[i]]
        int Cy = hy[y[i]]
        c = c + (Cx-1) * (Cy-1);
    }

    return c;
}
```

38 Given N distinct points, calculate no: of rectangles are formed such that sides are parallel to x-axis & y-axis

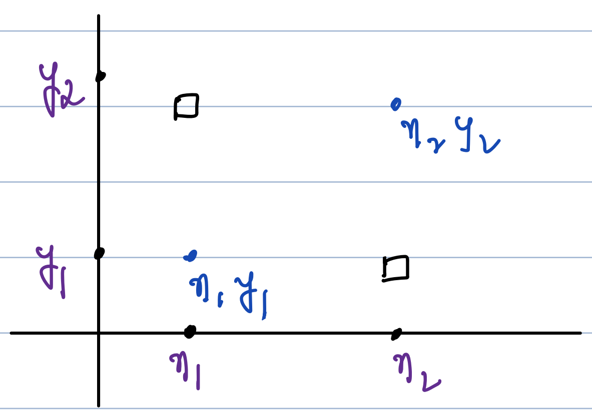
Expected TC: $O(N^2)$

Ex: N = 6 (3, 1) (1, 1) (3, 3) (1, 3) (3, 5) (1, 5)



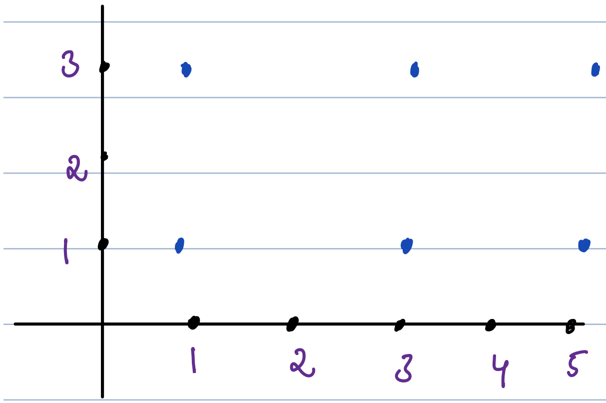
Idea1:

Idea2:



Dry Run:

	0	1	2	3	4	5
N = 6	(3, 1)	(1, 1)	(3, 3)	(1, 3)	(5, 1)	(5, 3)



Fin 2 points

x_1 y_1	x_2 y_2	x_2 y_1	x_1 y_2
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		
( )	( )		

```
int Rectangle(vector<int> &x, vector<int> &y){
```