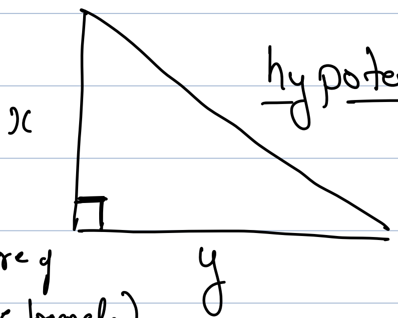
Right Angled Triangle Basics

→ one of the angles $= 90^\circ$

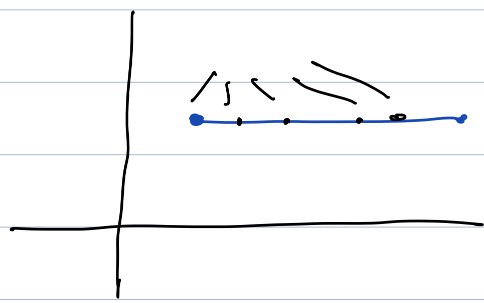


hypotenuse $\left(\sqrt{x^2+y^2}\right)$

→ longest side

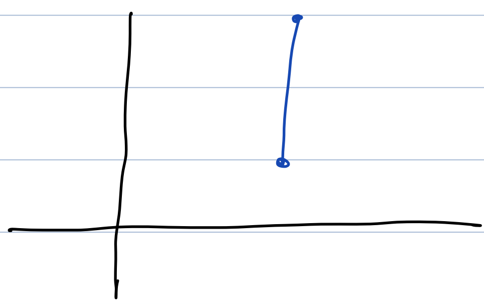
[Centre of the triangle)

Line parallel to x axis



⇒ y is same

Line parallel to y axis

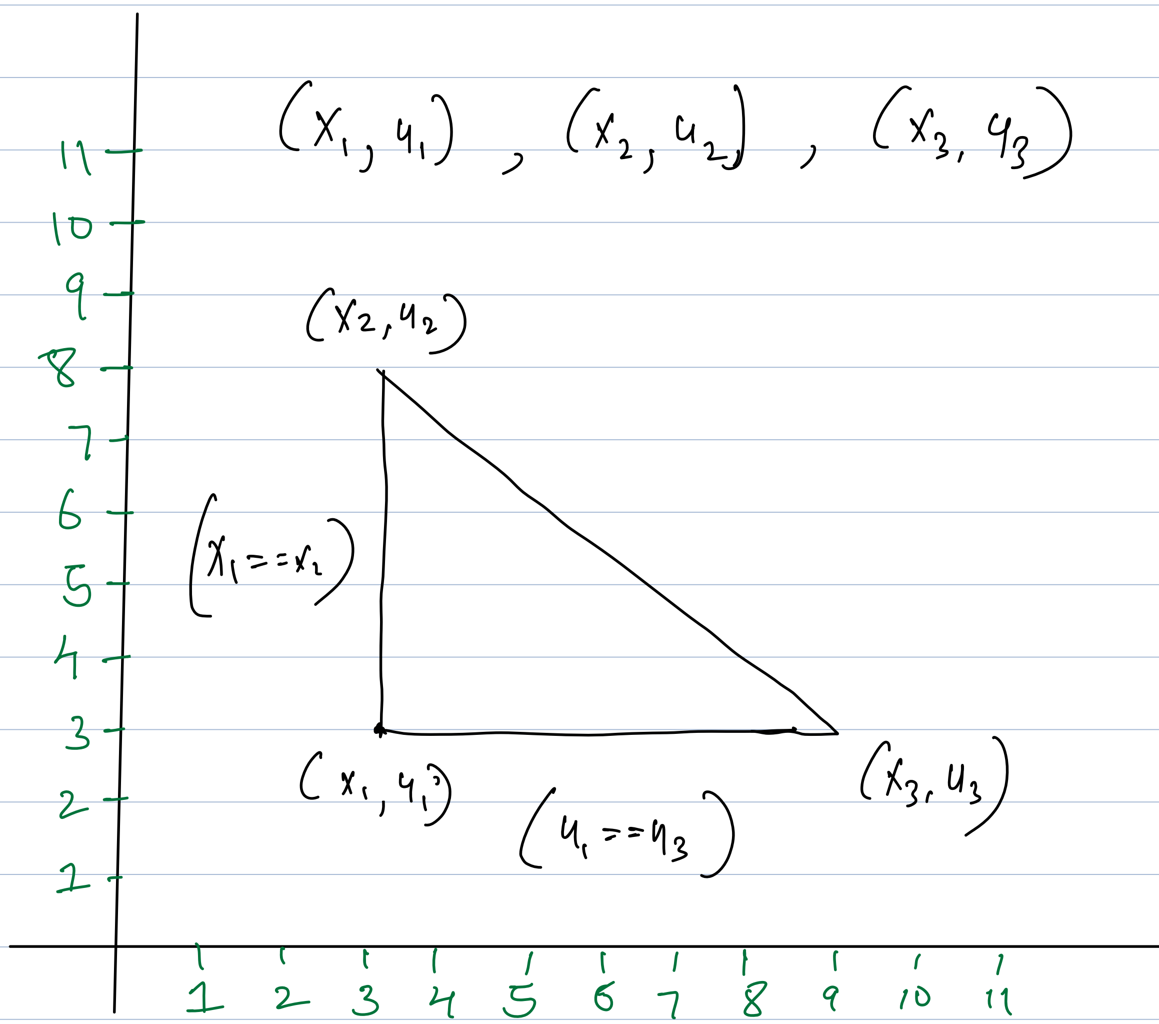


⇒ x is same

Q1) Given 3 distinct points in a 2D plane check. If they form right angled triangle [such that shorter sides are parallel to x-axis & y-axis respectively].

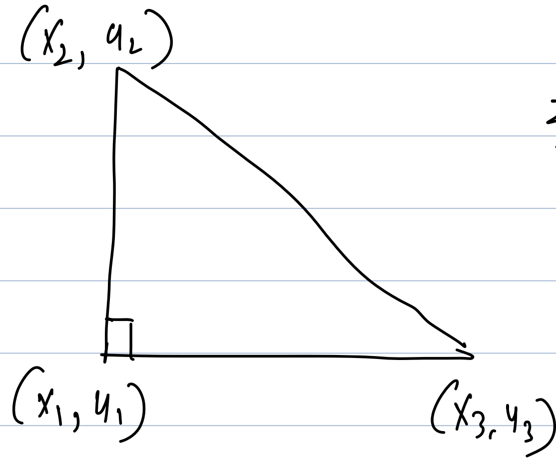
Ex1 : (1, 8) (1, 4) (5, 4) = YES
Ex2 : (5, 10) (1, 3) (5, 3) = YES
Ex3 : (4, 3) (8, 3) (4, 8) = YES

(x_1, y_1), (x_2, y_2), (x_3, y_3)

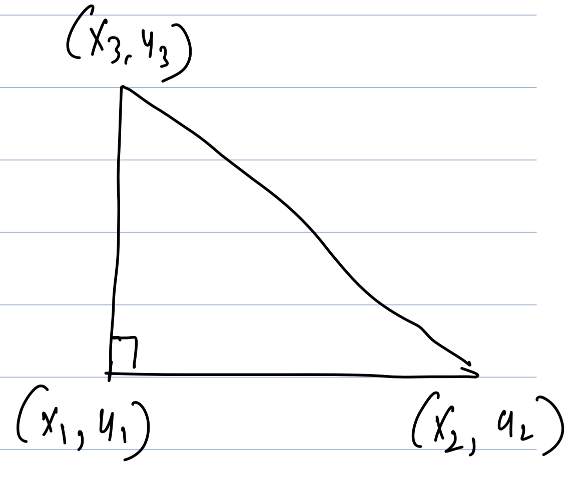


6 conditions

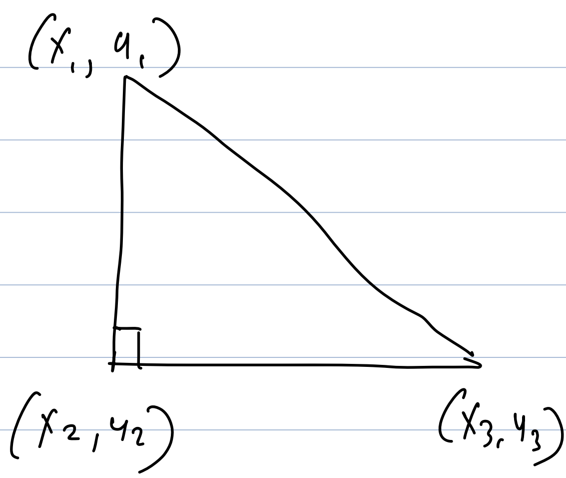
1)



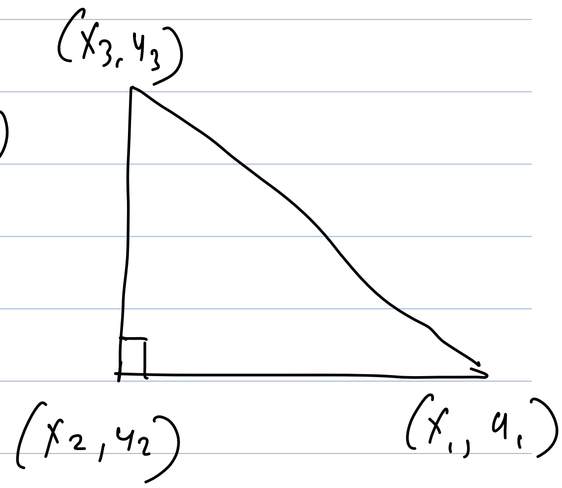
2)



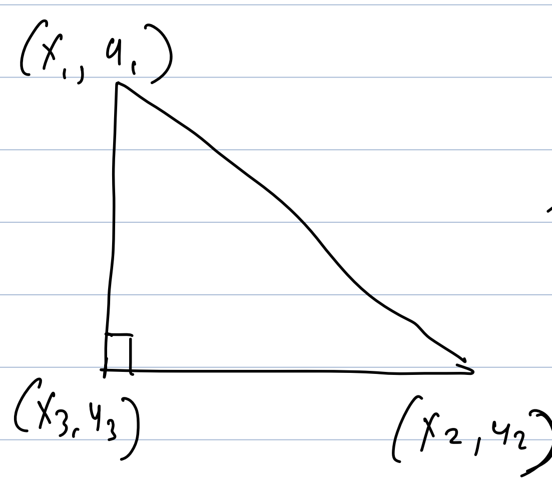
3)



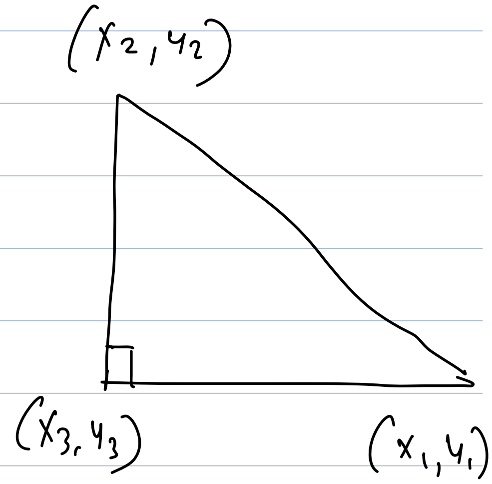
4)



5)



6)



Pseudo Code

Input : (x_1, y_1), (x_2, y_2), (x_3, y_3)

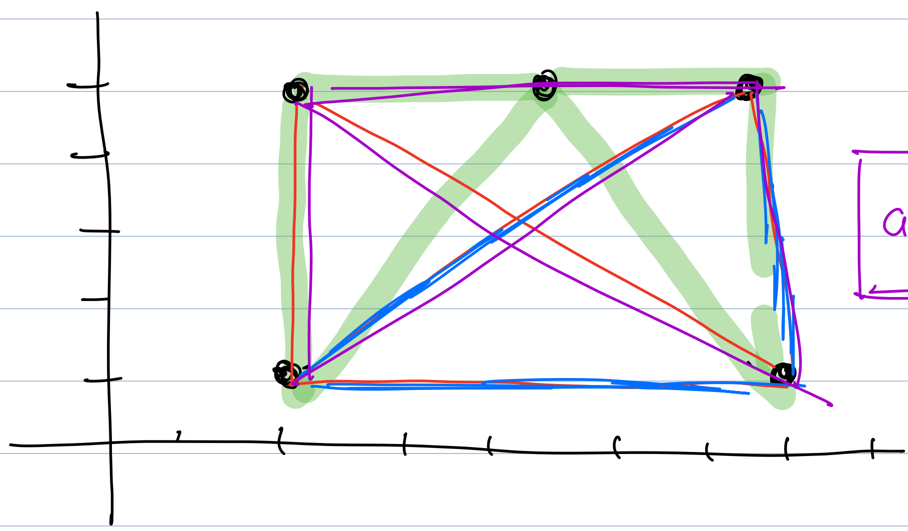
```
bool isRightAngleTriangle (x1, y1, x2, y2, x3, y3) {
    if ( x1 == x3 && y1 == y2) {
        return true;
    } else if ( x1 == x2 && y1 == y3) {
        return true;
    } else if ( x2 == x3 && y2 == y1) {
        return true;
    } else if (x2 == x1 && y2 == y3) {
        return true;
    } else if (x3 == x1 && y3 == y2 ) {
        return true;
    } else if (x3 == x2 && y3 == y1) {
        return true;
    } else
        return false;
}
```

Tc : O(1)

Sc : O(1)

Q2) Given N distinct points in a 2D plane, calculate no of $\triangle$ triangles that can be formed such that shorter sides are parallel to x-axis & y-axis.

Note: Input will be in form of 2 arrays x[n] & y[n], where i^{th} coordinate = (x[i], y[i])

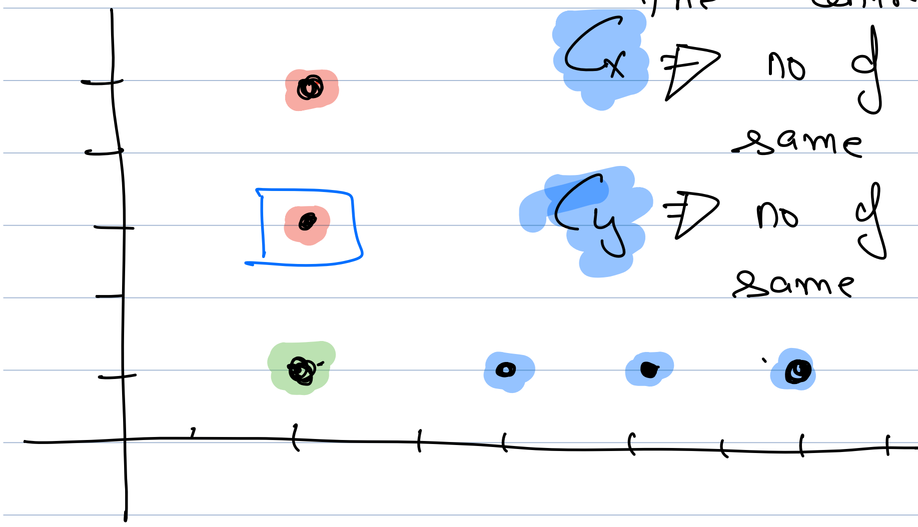


$\boxed{\text{ans} = 6}$

Brute Force: For all triplets, pass it through the code for the last Question

TC: $O(n^3)$

Assuming Green point to be the centre.



$C_x \Rightarrow$ no of points having same x axis

$C_y \Rightarrow$ no of points having same y axis.

Total $\triangle$ triangles having Green point as centre.

$$\Rightarrow (C_x - 1) \times (C_y - 1)$$

Note: There will be no duplicate right angled triangle as the centers would be different.

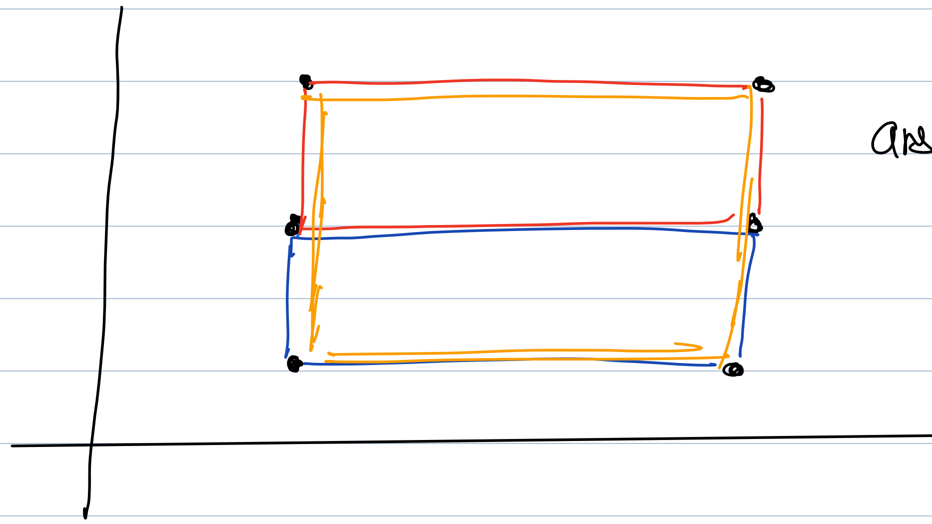
Pseudo Code

```
Hashmap<int, int> x_coord;
Hashmap <int, int> y_coord;

for (int i=0 ; i<n ; i++) {
        x_coord [x[i]]++;
        y_coord [y[i]]++;
}

int ans => 0;

for (int i=0 ; i<n ; i++) {

    int c_x => x_coord [x[i]];
    int c_y => y_coord [y[i]];

    ans = ans + (c_x - 1) x (c_y - 1);

}

return ans;
```

TC : O(n)
SC : O(n)

Q3 Given N distinct points on a 2D plane. Find no of rectangles such that their sides are parallel to x-axis & y-axis.

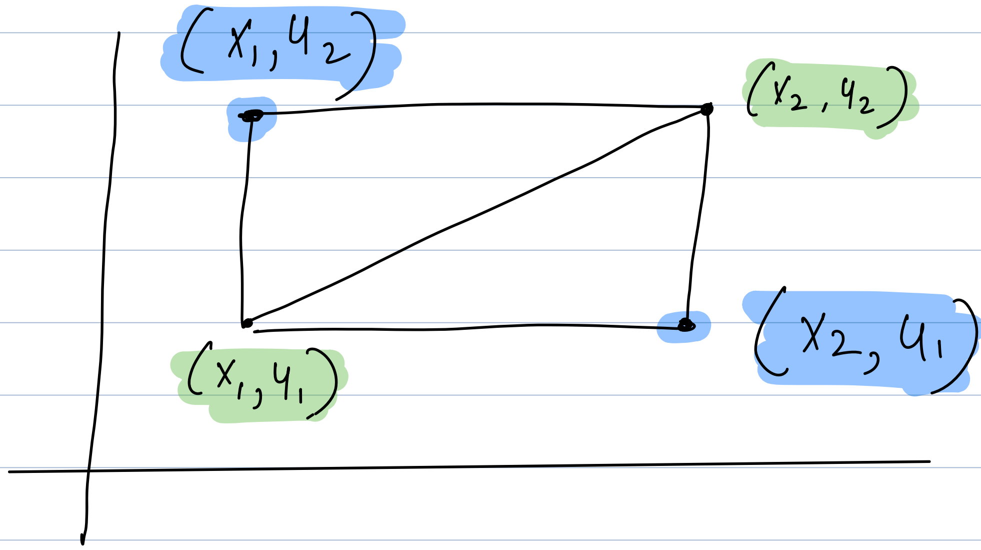
Ex1



ans ⇒ 3

Brute Force: for all 4 points

TC: $O(n^4)$



1) We will store a pair in Set

class Pair {
    int x;
    int y;
}

unordered_set → O(1)
ordered_set → O(logn)

unordered map → O(1)
ordered map → O(logn)

unordered_set → O(1)

Int, String, float, double, ~~Object~~, ~~Pair~~

ordered_set ⇒ Can take complex keys.

Pseudo Code

```
ordered_set < Pair <int, int>> Ts;
   for (int i=0 ; i<n ; i++) {
        Ts.insert ( make_pair (x[i], y[i] ) );
   }
        int ans => 0;
   for (int i=0 ; i<n ; i++) {
      for (int j=i+1 ; j<n ; j++) {

          x1 , y1 => x[i], y[i]
          x2 , y2 => x[j], y[j]
          if (x1 == x2  || y1 == y2)
                    continue
   Pair <int, int> first.point = make_pair (x1, y2);
   Pair <int, int> second.point = make pair (x2 , y1);

   if (Ts.contains (first.point) && Ts.contains (second point)){
          ans ++;
   }
}
        return ans/2;
```

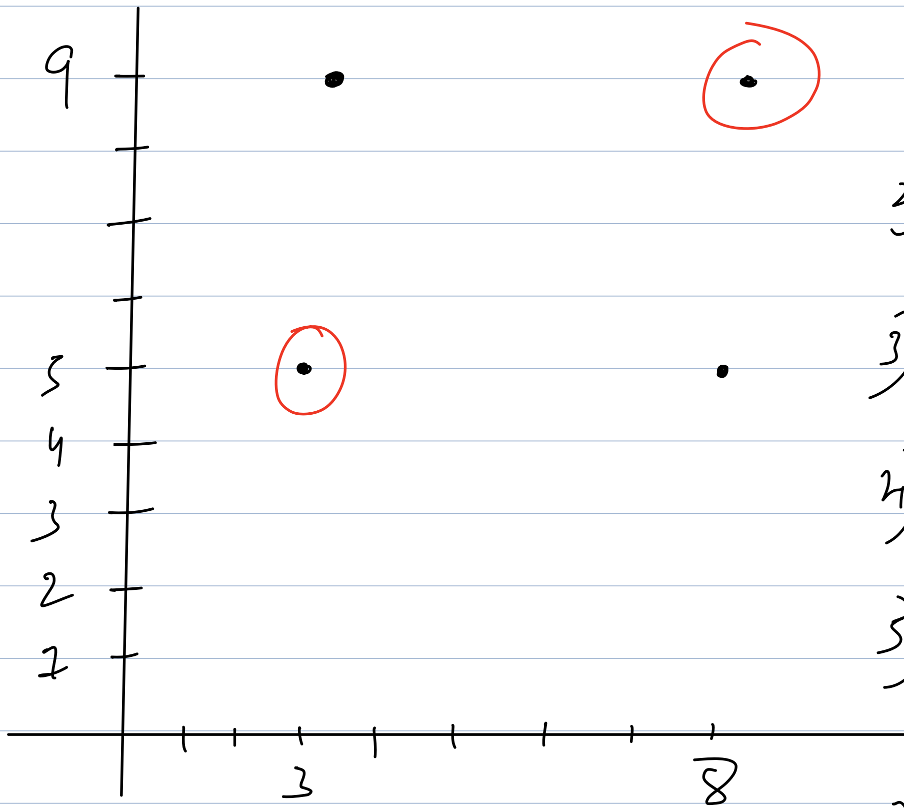
$Tc: n^2 \log n$

$Sc: O(n)$

Example

$x[\,] = \{3, 3, 8, 8\}$

$y[\,] = \{9, 5, 9, 5\}$ $\Rightarrow$ $(3,9)$, $(3,5)$, $(8,9)$, $(8,5)$



1) $(3,9)$ $(3,5)$ ✗
2) $(3,9)$ $(8,9)$ ✗
3) $(3,9)$ $(8,5)$ ✓
4) $(3,5)$ $(8,9)$ ✓
5) $(3,5)$ $(8,5)$ ✗
6) $(8,9)$ $(8,5)$ ✗

Total ⇒ 168
Sleep ⇒ 49 (7 x 7)
Office ⇒ 45 (9 x 5)
normal chores ⇒ 28 (4 x 7)
Scaler Class ⇒ 7.5 (2.5 x 3)
Scaler revise ⇒ 15 (5 x 3)

20 hours

⇒ 145

backlog.

1 Week off from office

Sat - Sun ⇒