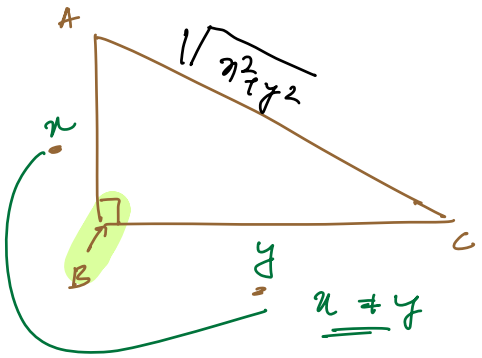
Today's Content:

- Given N points No: of triangles can be formed
- Given N points No: of rectangles can be formed
- Given a Text T, how many occurence of pattern P is present

1) Right Angled Triangle Basics

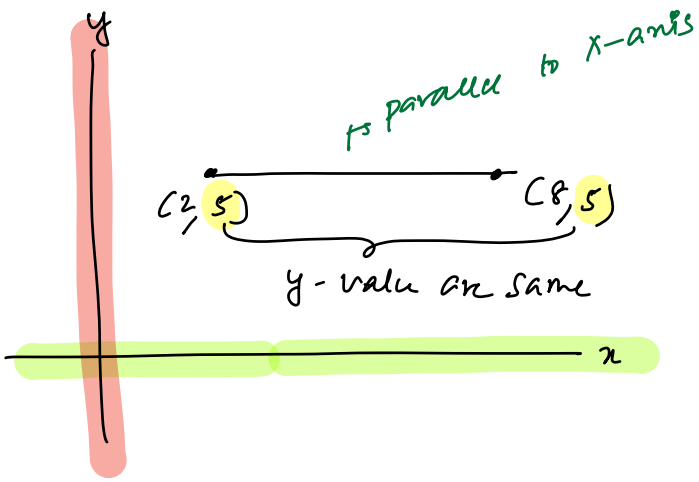
↳ one of degree = 90°

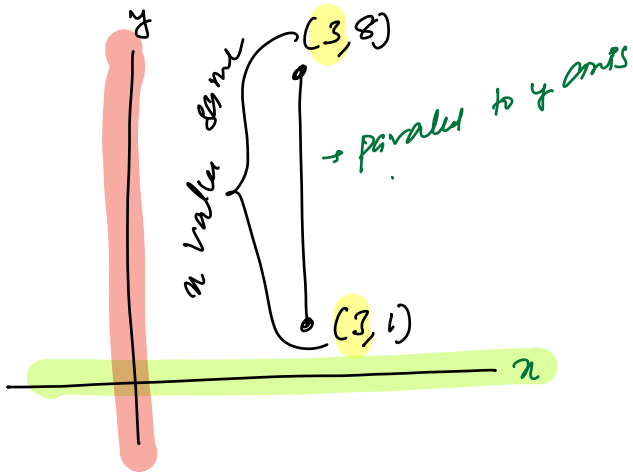


$\overline{AC}$ is longest side

Hypotenuse is your longest side

2) Parallel to X-axis / Parallel to Y-axis





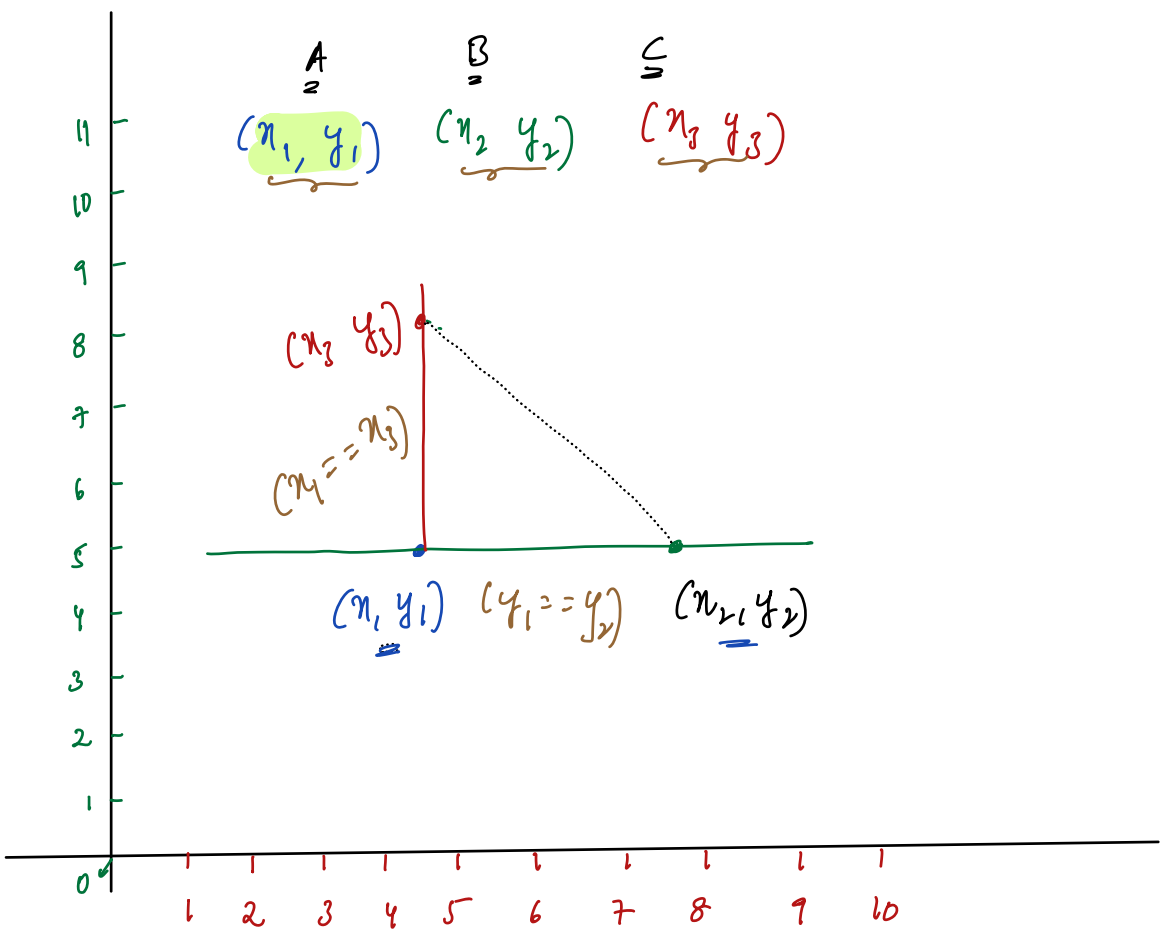
28) Given 3 distinct points in a 2D plane check if they form triangle, [such that shorter sides are parallel to x-axis & y-axis respectively] obs: Right angled Triangle other than hypotenuse

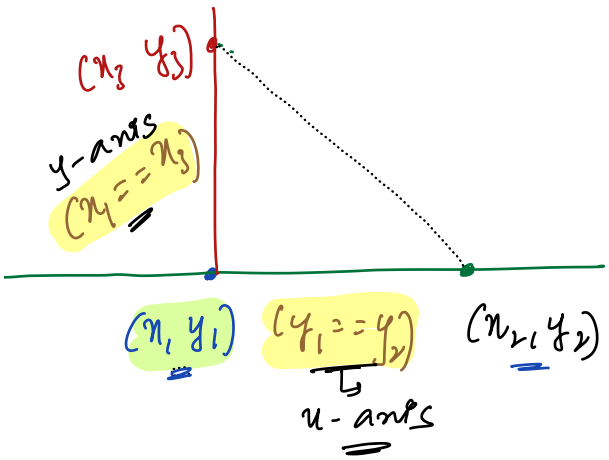
A B C

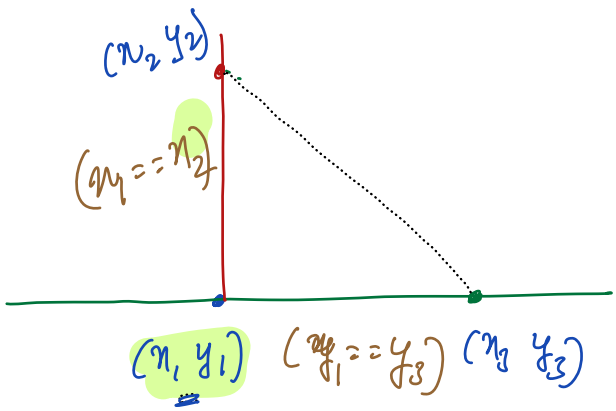
Ex1: (1, 8) (1, 4) (5, 4)

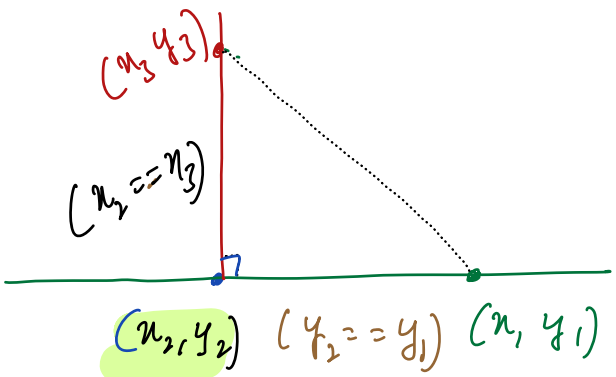
Ex2: (5, 10) (1, 3) (5, 3)

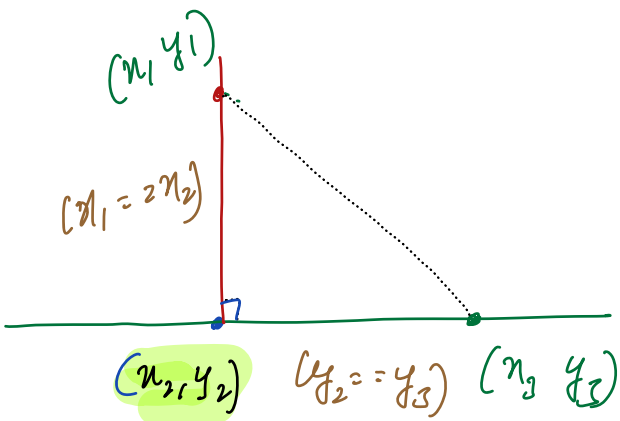
Ex3: (4, 3) (8, 3) (4, 8)

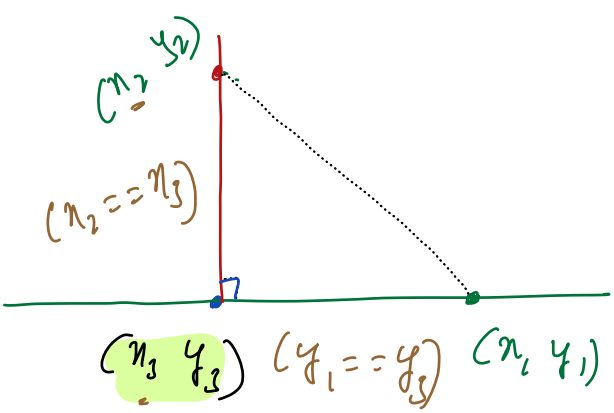


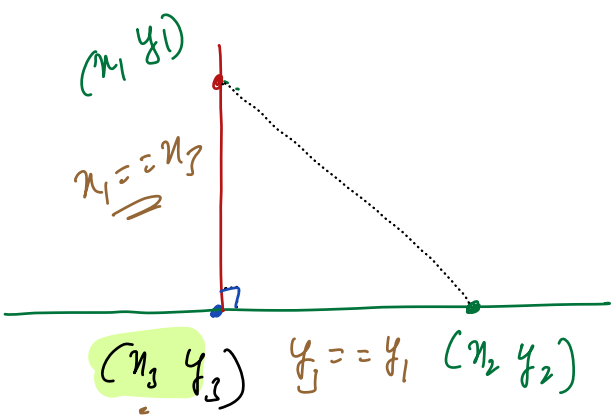










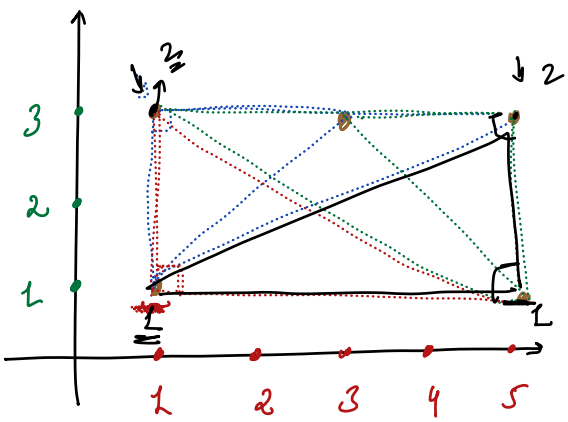


4 Pseudocode : (TODO)

35) Given N distinct points in a 2D plane, calculate no:- of triangles are formed such that shorter sides are parallel to x-axis & y-axis

Note: 2 Arrays x[N], y[N] are given. such that $i^{th} = (x[i], y[i])$

Eg:



ans = 6

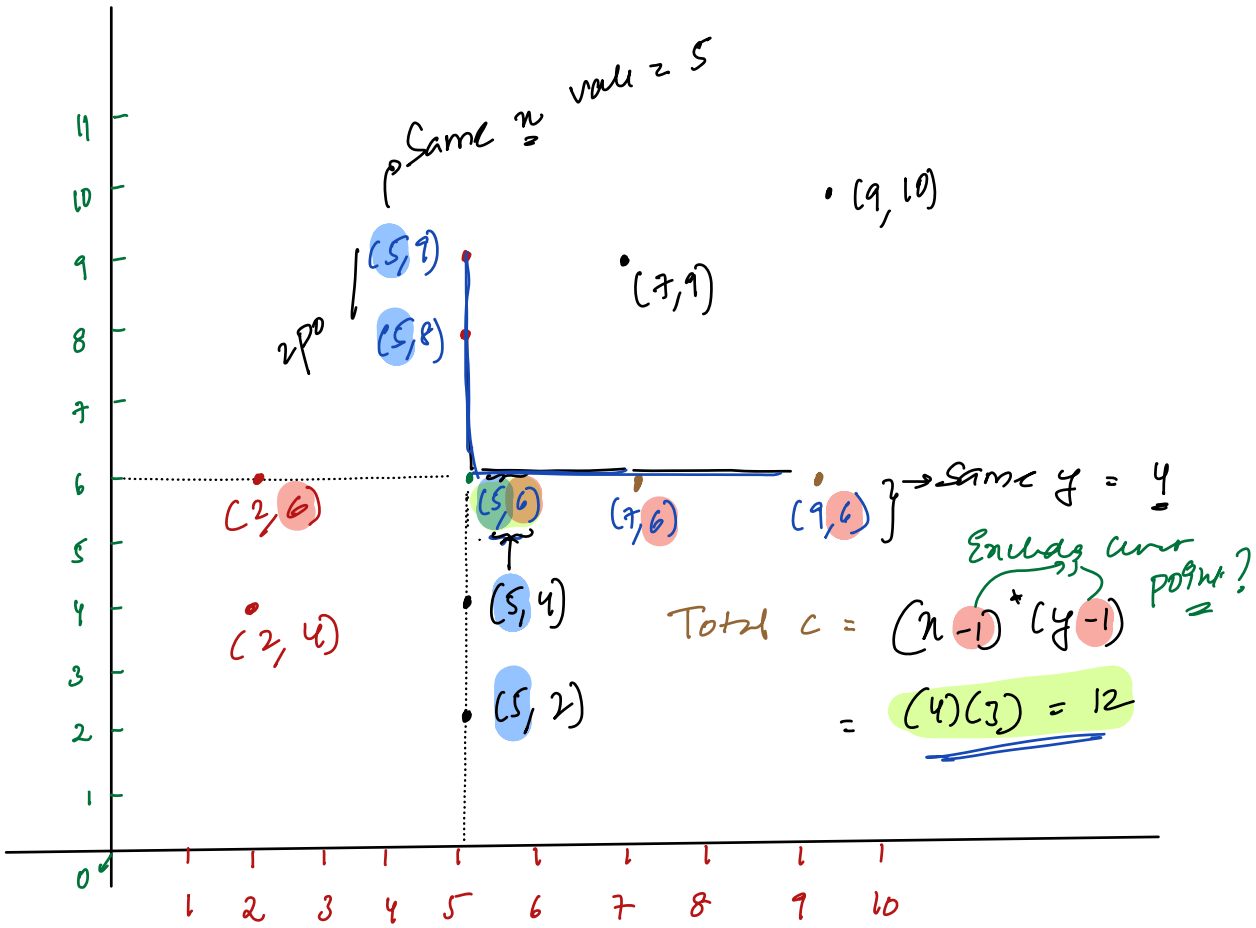
BF: for every tripler check if they are forming 90° Triangle, to x-axis, y-axis

TC: O(N3)

SC: O(1)

Hint:

Eg1:



PseudoCode

```
hashmap<int, int> hmx, hmy
Insert all (x) points in hmx & frequencies
Insert all (y) points in hm & frequencies
```

TC: O(N)

Take every point (x[i], y[i]) as 90° vertex

```
count number of points having x = x[i], hmx[x[i]]
count number of points have y = y[i], hmy[y[i]]
ans = ans + (hmx[x[i]] - 1) * (hmy[y[i] - 1])
```

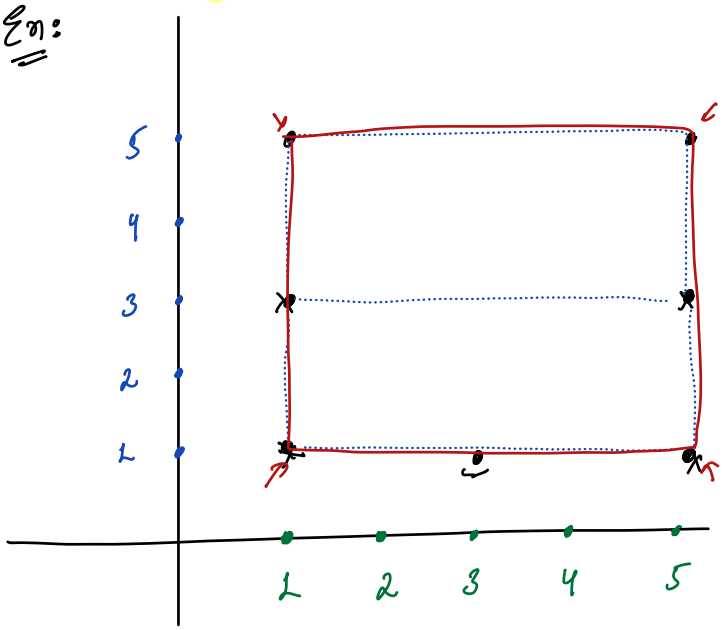
SC: O(N)

PseudoCode :

Given N distinct points in 2D plane, find no: of rectangles such that their sides are parallel to x-axis & y-axis

Note: 4 points should be at 4-corners

Ex:



ans = 3

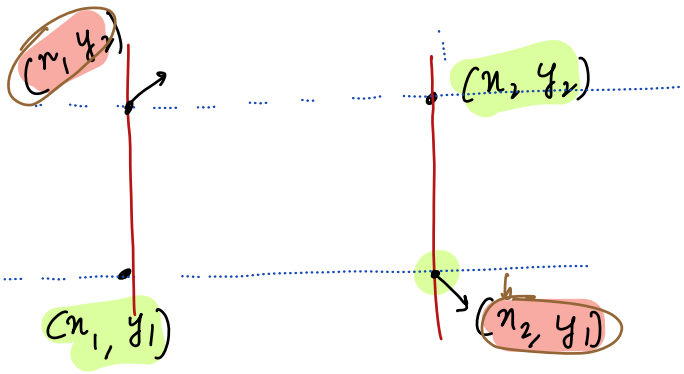
BF Code:

- for every points check
  - Rectangle, parallel to x-axis & y-axis

TC: $O(N^4)$ SC: $O(1)$

Hint : Quiz :

Say we fix 2 opp diagonal points in rectangle



↳ If (x_2, y_1) & (x_1, y_2) are also present in points

can I say rectangle formed.

Note: In your HashSet / HashMap key → Single Value

Implemented my Trees BBST not as HashFunction

Note: In your Treeset / Treemap : key → Any type → leaf → pair → Sort

TC for each operation → ($\log_2 N$)

library in your language of choice → Java

Treeset: <k>
- insert(k)
- delete(k)
- search(k)

Treemap: <key, value>
- insert(k, v)
- update(k)
- delete(k)
- search(k)

Advanced:
Data is ordered based on keys

If data is sorted on keys?
→ insert all data treemap / treeset
→ If we iterate treemap / treeset & print all keys → {sorted order}

→ TC → $O(N \log N)$

SC → $O(N)$

Ex: Treemap <String, int> Tm

	keys	Val
Tm. insert	<India,	800>
	<China,	500>
	<Nepal,	100>
	<America,	1000>

→ print

America	1000
China	500
India	800
Nepal	100

[ordered based on keys]

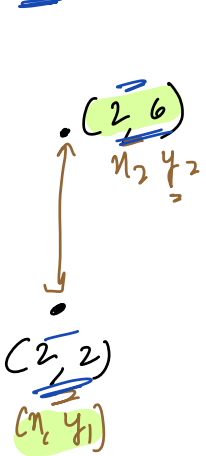
Pseudo code

↳ we are creating a Treeset

```
TreeSet < pair<int, int> > TS;

for(i=0; i<N; i++){ // Insert all points in Treeset
    TS.insert( make_pair{ x[i], y[i] } )
}

for(i=0; i<N; i++){
    (x1, y1) = (x[i], y[i])
    for(j=i+1; j<N; j++){
        (x1, y1) = (x[i], y[i])   (x2, y2) = (x[j], y[j])
        Edge Case -2
        if( x1 == x2 || y1 == y2 ){ // Rectangle not formed
            continue;
        }
        if( [(x2, y1) is in TS] && [(x1, y2) is in TS] ){
            c = c+1
            // we have a rectangle
        }
    }
}
return c/2; // Edge Case -1
```



⇒ Even, they are not diagnol

⇒ pairs = {{3, 9} {3, 5} {8, 9} {8, 5}}

Pair:

(3, 9) • (8, 9)

(3, 5) • (8, 5)

{3, 9} {3, 5} * x same

{3, 9} {8, 9} * y same

{3, 9} {8, 5} = 1

{3, 5} {8, 9} = 1

{3, 5} {8, 5} * y same

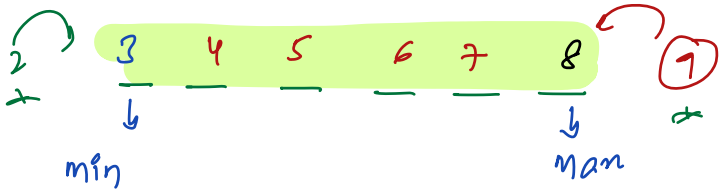
{8, 9} {8, 5} * x same

same vertex we count 2 times ← 2

// C ⟶)

Doubts: → Dsa

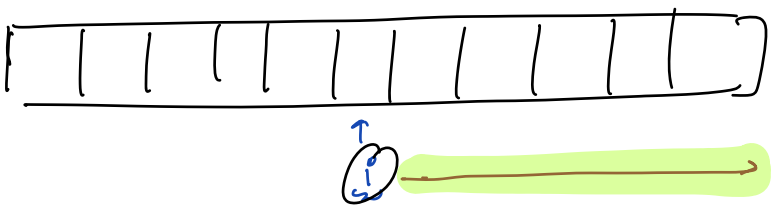
3 6 8 4 5 7



array → 6

min → 3
max → 8
→ 8 - 3 + 1 == 6

// Min subarray sum with atleast k. (+ve/-ve)



1) Pf[j] - Pf[i] >= k → Pf[j] >= (k + P[i])
2) j >= i → j → [i+1, N-1]

{ Segment Tree } //

→ { }

Deque: For every window k get max Elem

HashMap<Int, Int>
key = Val

<2, 3>
<2, 8>

TreeMap<Int, Int>
key = Value