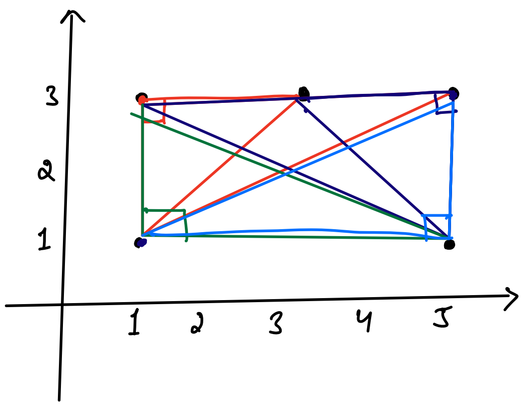
Agenda :-

→ Questions on Hashing

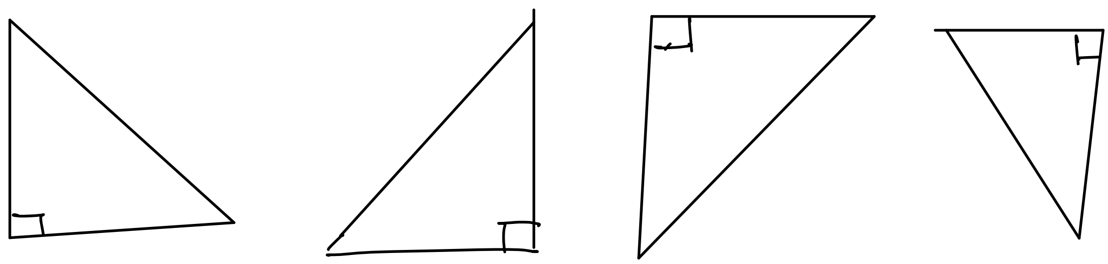
→ Rolling Hash

→ Rabin Karp Algorithm

Q) Given co-ordinates of N distinct points on a 2D. plane. Count the no. of right angled triangle using the given set of points such that two small sides (P and B) of a △ should be parallel to x-axis & y-axis.



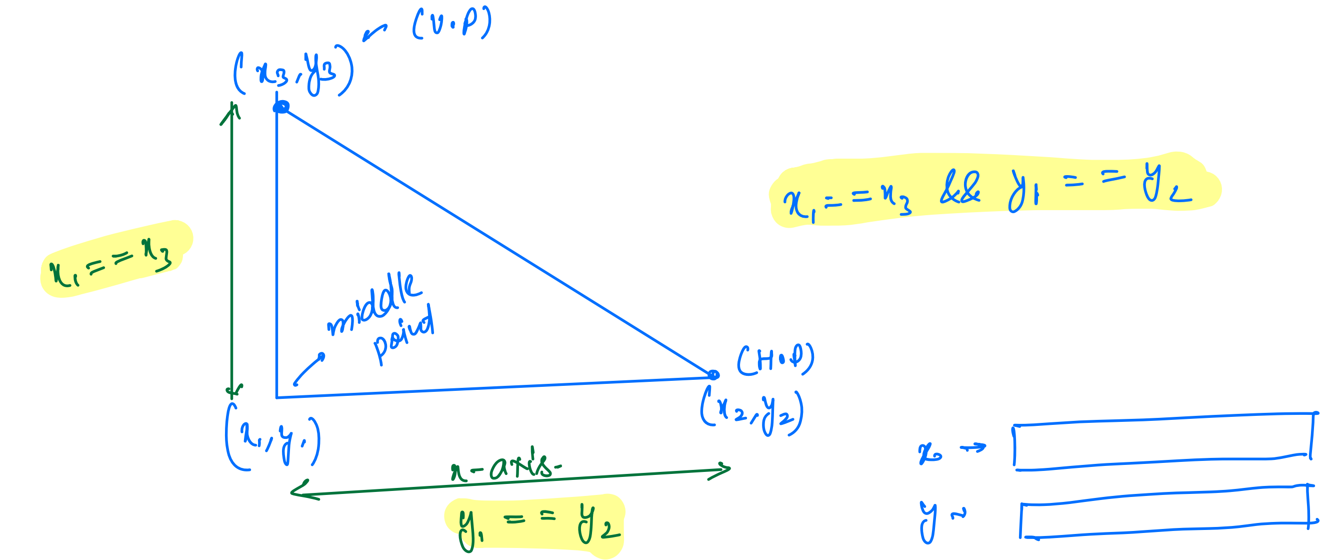
ans=6.



triangle → triplet of co-ordinates is required.

idea-1. → Consider all the triplets & check if they can form a right angled triangle or not.

two-sides → { ∥ x axis
∥ y axis

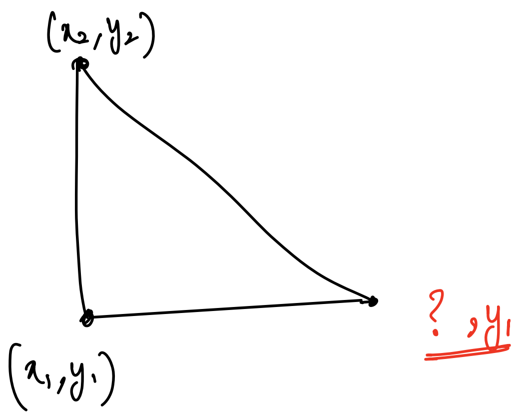


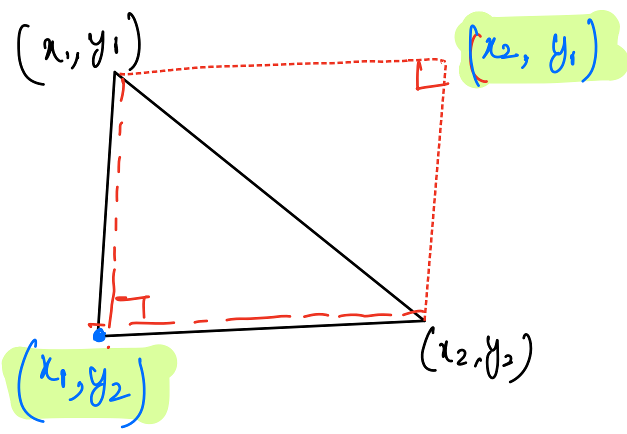
// All triplets.

```
for( i=0; i < N; i++){  // i → middle point
    for( j=0; j<N; j++){  // j → vertical point
        if(i==j) { continue }
        for( K=0; K<N; K++){  // K → horizontal point
            if(K==i || K==j) { continue }
            if( x[i] == x[j] && y[i]==y[K]){
                count ++
            }
        }
    }
}
return count;
```

$T.C \to O(N^3)$
$S.C \to O(1)$

$O(N^2)$?





idea. → using Hashset / HashMaps.

// Add all the points in the hashset.

```
for( i = 0 ; i < N ; i++) {
    for( j = i + 1 ; j < N ; j++) {
        //handle edge cases.
        check ( x[i], y[j]) { count ++ }
        check( x[j], y[i]) { count ++ }
    }
}
return count ;
```

T.C → O(1).

T.C → O(N^2)
S.C → O(N)

O(N)

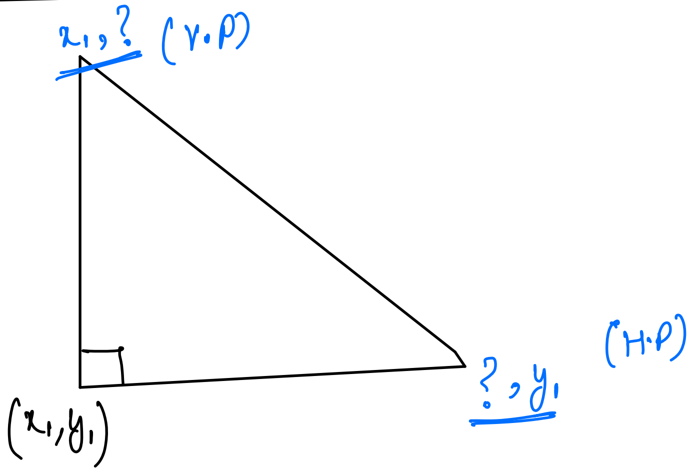
x →	2	4	4	3	4
	0	1	2	3	4
y →	5	3	1	7	7
	0	1	2	3	4

[2-5 , 2-3, 4-1 , 3-7 , 4-7]

Key → string → x_y

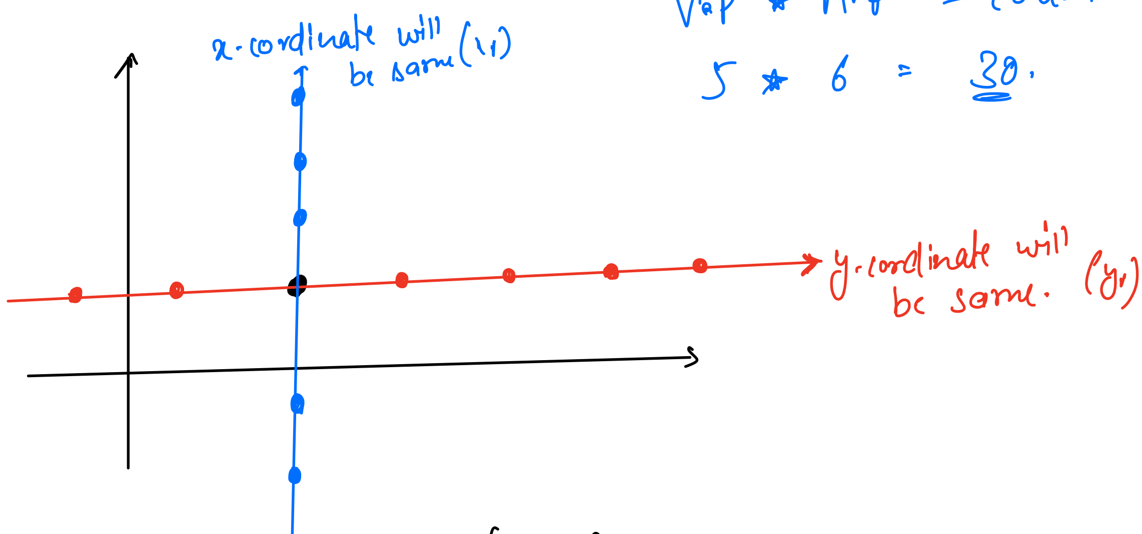
Syntax.

hs.contains(x[i] + "_" y[j]) { count ++ }



V.P * H.P = count.

5 * 6 = 30.

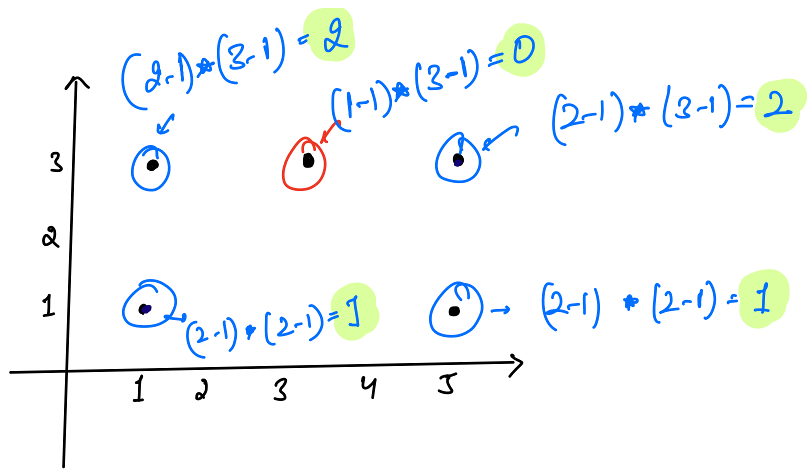


(x_1, y_1)

{ Count of co-ordinates with x-cordinate as x_1 - 1 } * { Count of co-ordinates with y-cordinate as y_1 - 1 }

[2 frequency HashMaps.

x-cordinates   y-cordinates]



ans = 6.

// pseudo-code.

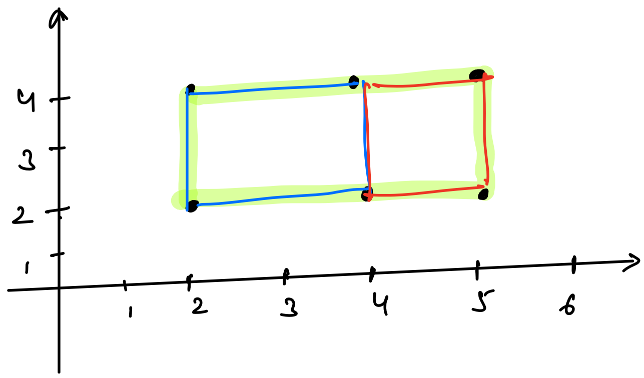
① Take 2 hashmap. hm1 (x) and hm2 (y)

② Insert all x-cordinates in hm1 & all y-coordinates in hm2.

③
```
for (i = 0 ; i < N ; i++) {
        c1 = hm1[x[i]] - 1
        c2 = hm2[y[i]] - 1
        ans += (c1 * c2) ;
}
return ans;
```

$\boxed{\begin{array}{l} T.C \rightarrow O(N) \\ S.C \rightarrow O(N) \end{array}}$

Q. Given N points on a 2-D plane. Find count of rectangles we can form such that sides are parallel to x-axis and y-axis.

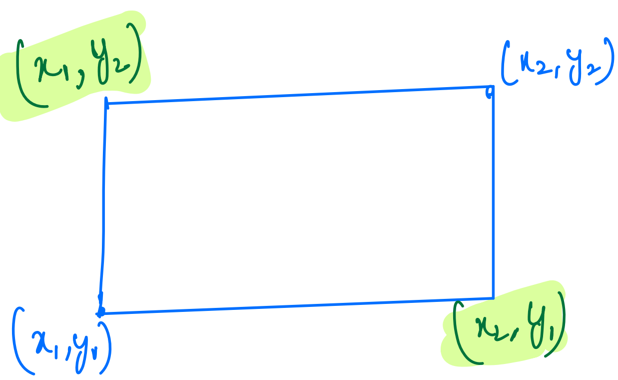


[ans = 3.]

idea.1 - Consider all the quadruplets & check if they are satisfying the condition.

T.C → $O(N^4)$

(N^3 ✓) (N^2 ✓)



⇒ { # todo }

Text : S : a b c b a x b a (N)

pattern : T : c b a (K)

⇓

Check if pattern is present as a substring in text ?

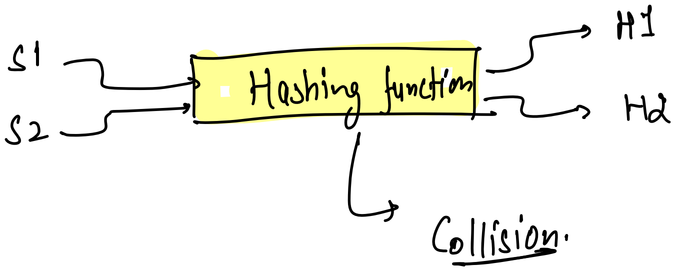
idea-1 Consider all the sub-string of size K (length of pattern)

$T.C \rightarrow O(N^2)$

{ KMP
Z-Algo
Rabin Karp }

Hashing?

integer



.H1 == H2

① Sum of ASCII value.

a b c
↓ ↓ ↓
0 1 2 ⇒ 3.

{ a b c
a c b
b a c
b c a
c a b
c b a }

{ lot of collision will be there. }

② ASCII values with indices.

$$\sum ch(i) * (i+1)$$

a b c
0 1 2

$\Rightarrow (97*1) + (98*2) + (99*3)$

$\Rightarrow$ collision.

③ Decimal no.

4 1 3 9 6 2

$\Downarrow$

$$4 \times 10^5 + 1 \times 10^4 + 3 \times 10^3 + 9 \times 10^2 + 6 \times 10^1 + 2 \times 10^0$$

$\boxed{0-9}$

b a d e f
0 1 2 3 4

$$\left[b * 26^4 + a * 26^3 + d * 26^2 + e * 26^1 + f * 26^0 \right]$$

$$\boxed{\sum s[i] * p^{N-i-1}}$$

$p = 26.$

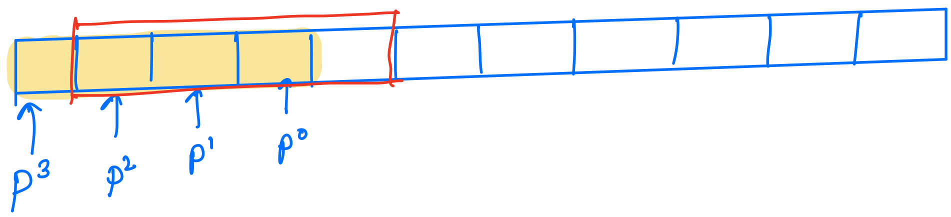
$\Downarrow$

very large.
{ overflow }

Polynomial Rolling Hash

$$\boxed{\sum s[i] * p^{N-i-1} \% m}$$

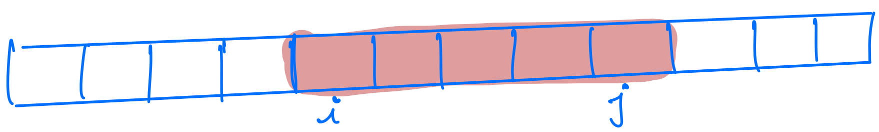
$[m \approx 10^9.]$



$H_1 = s[0] * p^3 + s[1] * p^2 + s[2] * p^1 + s[3] * p^0$

$H_2 = s[1] * p^3 + s[2] * p^2 + s[3] * p^1 + s[4] * p^0$

$\boxed{H_2 = (H_1 - s[0] * p^3) * p + S[4]} \implies T.C \to O(1)$



$H_{old} =$ i-1 to j-1

$H_{new} =$ i to j = ?

$$\boxed{H_{new} = (H_{old} - s[i-1] * p^{N-1}) * p + s[j]}$$

$\boxed{T.C \to O(N) \quad S.C \to O(1)} \implies$ Rabin Karp Algorithm uses Rolling Hash. (99% accurate)

{Plagrarism checker.}

Disadvantage? ⇒ collision

$$\boxed{\sum s[i] * p^{N-i-1} \% m} \Rightarrow \underline{0 \text{ to } m-1}$$

$m = 10^9$

H_1

0 — — — — — — — — — — — — $N-1$

$$S_1 = H_1 = 0$$

$$S_2 = H_2 = 1/m \Rightarrow 10^{-9}$$

$$S_3 = H_3 = 2/m$$

$$S_4 = H_3 = 3/m$$

$$\vdots$$

$$S_N = H_N = \frac{N-1}{m}$$

$N = 10^5$

$M = 10^9$

$$\frac{N}{m} = \frac{10^5}{10^9} = 10^{-4}$$

$$\left\{ \% \text{ of collision} = 0.0001 * 100 = 0.01\% \right\}$$