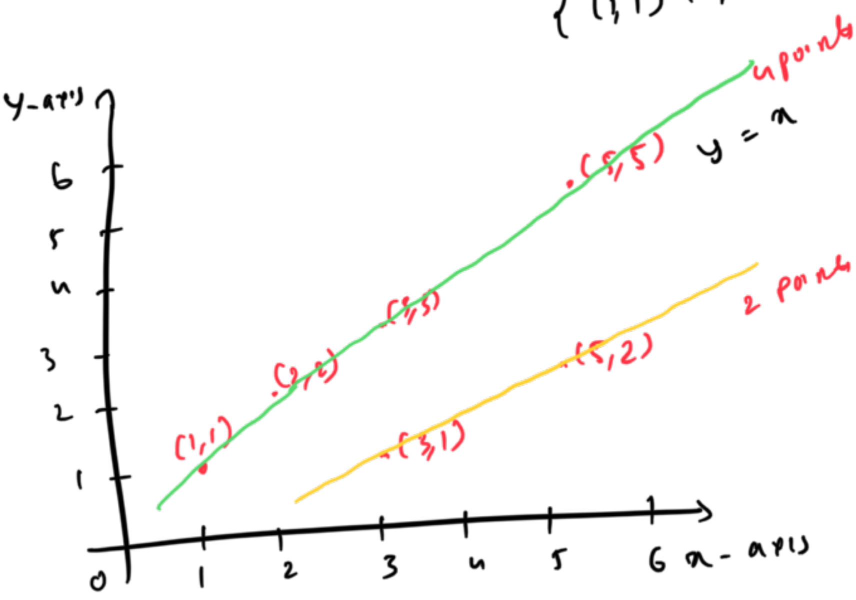
Hashing - 2

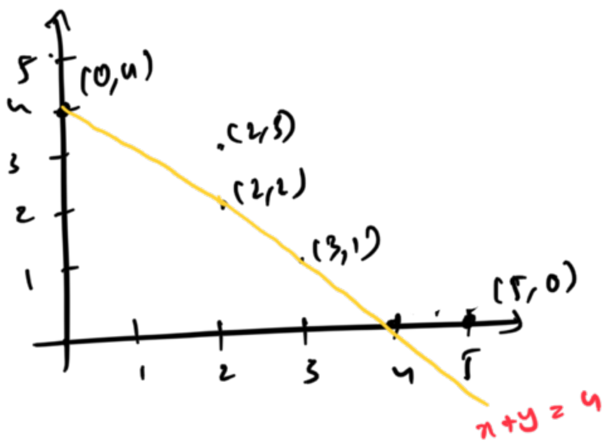
Question: Max No. of Points on the same line

Given N points on a 2D plane (x, y), find the max no. of points which lie on the same line

vector<pairs>
{ (1,1) (2,2) (3,1) (5,2) (5,5) }

Ex:





$y = -x + 4$

Ans = 3

Equation of line

1) $\boxed{y = mx + c}$

2) $ax + by = c$

{ m is slope
c is y-intercept }

3) $\frac{x}{a} + \frac{y}{b} = 1$

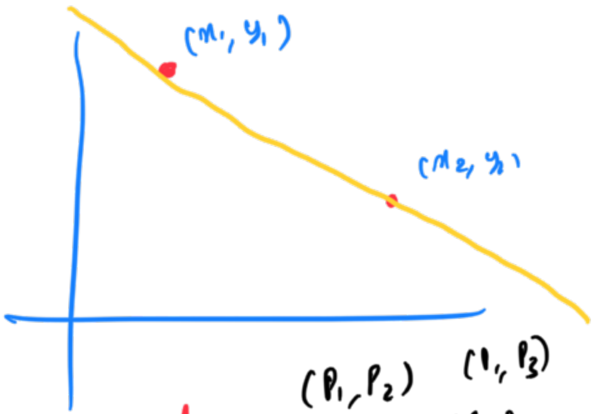
Brute Force :

consider all possibilities

Consider all lines $\Rightarrow$ select line with max no. of points

Min no. of Points required to form a line 2

(x_1, y_1) and (x_2, y_2)



$P1 = (x_1, y_1)$ ✓

$P2 : (x_2, y_2)$ ✓

(P_1, P_2) (P_1, P_3) (P_1, P_4)

(P_2, P_3) (P_2, P_4)

$\boxed{N_{C_2} \text{ Pairs}}$

Given N points

$P_1\,(x, y)$ $\quad P_2 \quad P_3 \quad . \quad . \quad . \quad . \quad P_n$

2) Consider all Pairs of Points $(n_{C_2}) = \frac{n(n-1)}{2}$ Pairs

(x_1, y_1) & (x_2, y_2)

$\rightarrow y = mx + c \rightarrow \{m, c\}$

$y_1 = m x_1 + c \quad$ — ①

$y_2 = m x_2 + c \quad$ — ②

② — ①

$y_2 - y_1 = m(x_2 - x_1) + \cancel{c} - c$

double $\boxed{m = \frac{y_2 - y_1}{x_2 - x_1}}$ — ①

$$y_1 = \frac{y_2 - y_1}{x_2 - x_1} \times x_1 + c$$

$$c = y_1 - \frac{x_1 y_2 - x_1 y_1}{x_2 - x_1} = \frac{x_2 y_1 - \cancel{x_1 y_1} - x_1 y_2 + \cancel{x_1 y_1}}{x_2 - x_1}$$

(x_i, y_i)

$y = mx + c$

double $\boxed{c = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}}$ — (2)

For one line

$O(n)$

```
for (i = 0; i < N; i++) {
    if (y[i] == m.x[i] + c)
        count++;
}
ans = max(ans, count)
```

(3.7831) (3.78332)

double

$O(n)$

Lines we are considering = $^nC_2 \rightarrow \frac{n(n-1)}{2}$

T.C: $O(n^2 \times n) = \boxed{O(n^3)}$

$\boxed{y = 3x + 5}$

n^2 : lines
n

$\boxed{O(n^3)}$

Issues:

1) $(5, 0)$ $(5, 2)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2 - 0}{5 - 5}$$

Run Time Err

2) Dealing with floating point numbers will have precision issues

$m = \frac{7}{3} = 2$

→ Use Only Integers

$$y_i = \frac{y_2 - y_1}{(x_2 - x_1)} \times x_i + \frac{y_1 x_2 - x_1 y_2}{(x_2 - x)}$$

$$\text{if}\left((x_2 - x_1) y_i == (y_2 - y_1) \times x_i + (y_1 x_2 - x_1 y_2) \right)$$

count++;

$$\boxed{\begin{array}{l} T.C: O(n^3) \\ S.C: O(1) \end{array}}$$

$y = 2x + 3$
$y = 2x + 5$

Approach 2:

store count of lines {m, c}

Find line eq => Finding {m, c}

P_1 P_2 P_3 . . . P_n

$\forall$ pair P_i, P_j, Find line equation {m, c}

Key: <m, c>

Hash{m, c}++

=> Answer is map Freq in hash map?

$\frac{K(K-1)}{2} = \boxed{F}$

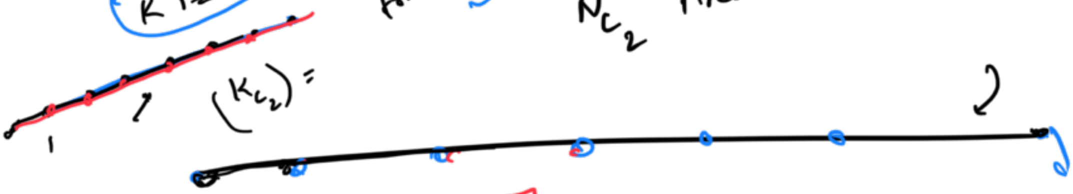
$K^2 - K = 2F$

$\boxed{K^2 - 2F - K = 0}$

K points

for {m, c} ++ N_{C_2} times

$({K_{C_2}}) =$



N points

$\boxed{\{m, c\}}$

$\boxed{n_{C_2} \text{ pairs}}$ => $\boxed{\text{hash}\{m, c\}\text{++}}$

(n_{C_2}) pairs

{m, c}++;

n_{C_2} pairs =>

$\boxed{\frac{n(n-1)}{2}}$ times

for(i=0 to n) for(j=i+1 to n) map[m,c]++;

$F = $

$(n-1)$

(i,j) =>

$$\max(F) = \frac{n(n-1)}{2} = {}^nC_2$$

$$2F = n^2 - n$$

$$\Rightarrow \boxed{n^2 - n - 2F = 0} \rightarrow \text{Quadratic}$$

$$n = \boxed{\frac{1 + \sqrt{1 + 8F}}{2}}$$

Integer

Roots

$$\boxed{ax^2 + bx + c \quad \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

```
for (i=0; i<n; i++) {
    for (j = i+1; j<n; j++) {
        (P1, P2)
        x[i], y[i] = P1
        x[j], y[j] = P2.
        // compute m, c
        map[<m,c>] ++;
    }
}
```

Find (m,c) with max Freq

Issues:

1) Precision Issues
{m, c} all double values

2) Infinite slope
pair<int, int>

$m = $ < num, den >

$\frac{2}{6}$ $\frac{3}{9}$

m = <2,6>   <3,9>

map[<2,6>]++
map[<3,9>]++

$c = \langle num, den \rangle$ map

$\langle 2, 6 \rangle \rightarrow \langle 1, 3 \rangle$

gcd(2,6) gcd(2,6)

$\frac{2}{6} \Rightarrow \frac{1}{3}$

$\frac{3}{9} \Rightarrow \frac{1}{3}$

$\langle \frac{2}{2}, \frac{6}{2} \rangle \Rightarrow \langle 1, 3 \rangle$

$\langle 3, 9 \rangle$

$gcd(3, 9) = 3$

$\langle \frac{3}{3}, \frac{9}{3} \rangle \Rightarrow \boxed{\langle 1, 3 \rangle}$

$\Rightarrow$ Do the same for y-intercept.

{m, c}

pair pair

Key: $\{ \{ \frac{num}{d}, \frac{den}{d} \}, \{ \frac{num'}{d'}, \frac{den'}{d'} \} \}$

m c

Mention the Hashfunction to convert integers

pair < pair, pair >

Map : O(log K) K is no. of entries in the hashmap

T.C : $n_{C_2} \times \log(X_i) \times \log n$

$\log(n^2) = \log n$

elements in Hashmap = $\binom{n}{2}$ n^2

$\boxed{S.C: O(n^2)}$

O(1)

$\boxed{T.C: n^2 \log n \cdot \log(X_i)}$

$m = \langle 4, 12 \rangle$

gcd (x, y)

log (min + ...)

$O(1)$: Unordered Map

Solution 3:

Don't want to compute c {y-intercept}

2 lines with same slope,

Line 1: m_1

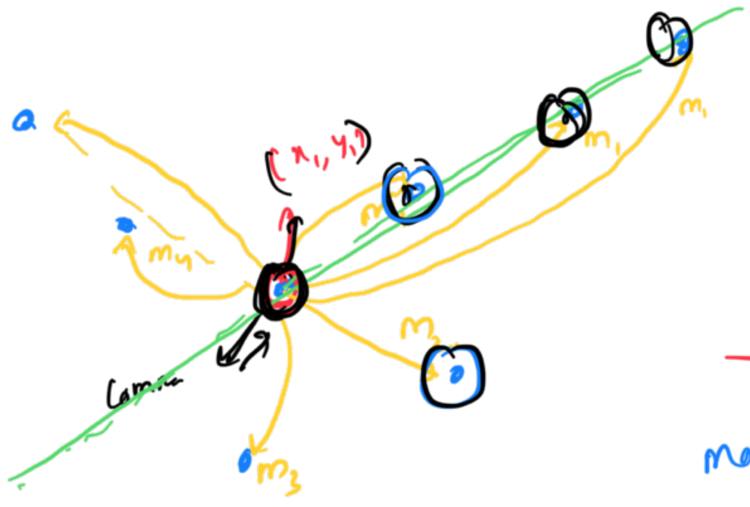
Line 2: m_2

$\boxed{m_1 == m_2.}$

2) If there is atleast one common point.

(parallel)

→ Fix one point, consider only passing through that point



N = 8

m_1 : 3 +1 = 4
m_2 : 1
m_3 : 1
m_4 : 2

max(map) + 1 ?

key in Hashmap:

$\boxed{S.C: O(n)}$

... is the answer?

T.C:

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

for(j=0; j<n; j++) {

// Find slope {m, d} → gcd(n, y)

map[{m, d}]++; ↙ log(n)

}

}

$\boxed{T.C: \; O(n^2 \times \underbrace{\log n}_{map} \times \underbrace{Cost(X_1)}_{gcd})}$

$P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5$

$(P_1, P_2) \quad (P_1, P_3) \quad (P_1, P_4) \quad (P_1, P_5)$

$(P_2, P_3) \quad (P_2, P_4) \quad (P_2, P_5)$

$(m_3, c_3) \quad (m_4, c_4) \quad (m_5, c_5)$

(m, c)

Hash = { ... }

n Keys

Hash = { (m1, c1) : 3, (m2, c2) : 4, (m3, c3) : 1, (m4, c4) : 2 }

max = F.

m = <num, den>

<3, 0> .1

$\boxed{<m, c>}$

{m, c}

Question: Count no. of Right angled triangles

Given N points on a 2D plane, find no. of right angled triangles such that

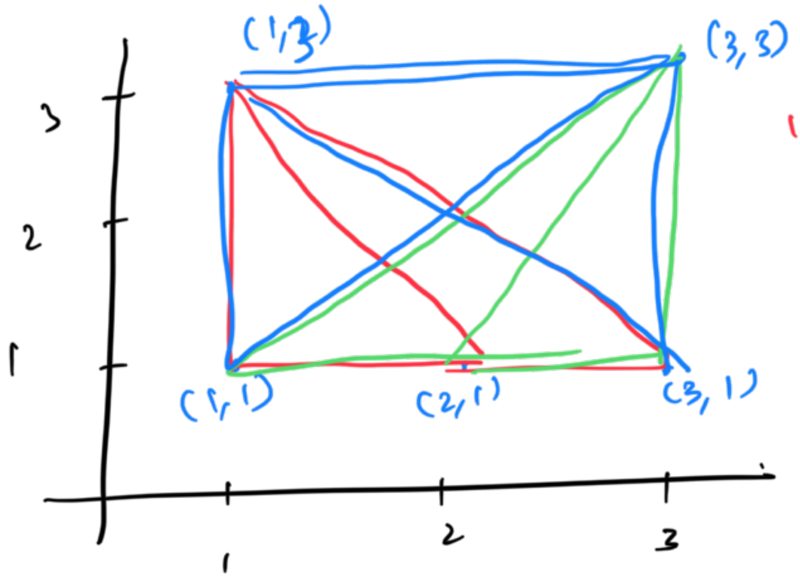
1) 1 side is parallel to x-axis

parallel to y-axis

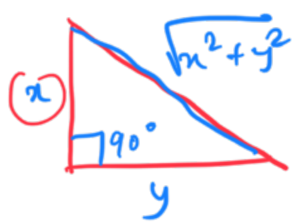
1) ... pa...

2) 1 side is

2 triangles are diff if they have atleast one different side



$1+1+1+1+1+1+1$
$= \boxed{6}$



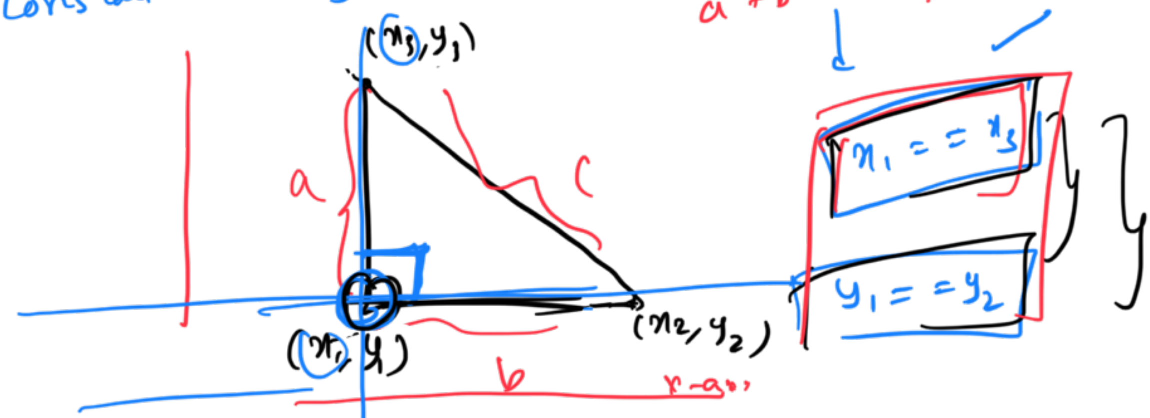
(Pythagoras theorem)

(P_1, P_2, P_3)
(P_1, P_3, P_2)
(P_2, P_1, P_3)

Brute force :

Consider every triplet (P_i, P_j, P_k)

$a^2+b^2=c^2$?



Given 3 points (x_1, y_1) (x_2, y_2) (x_3, y_3)

→ $\boxed{(x_1, y_1)}$ to be the hypotenuse point || to x-axis

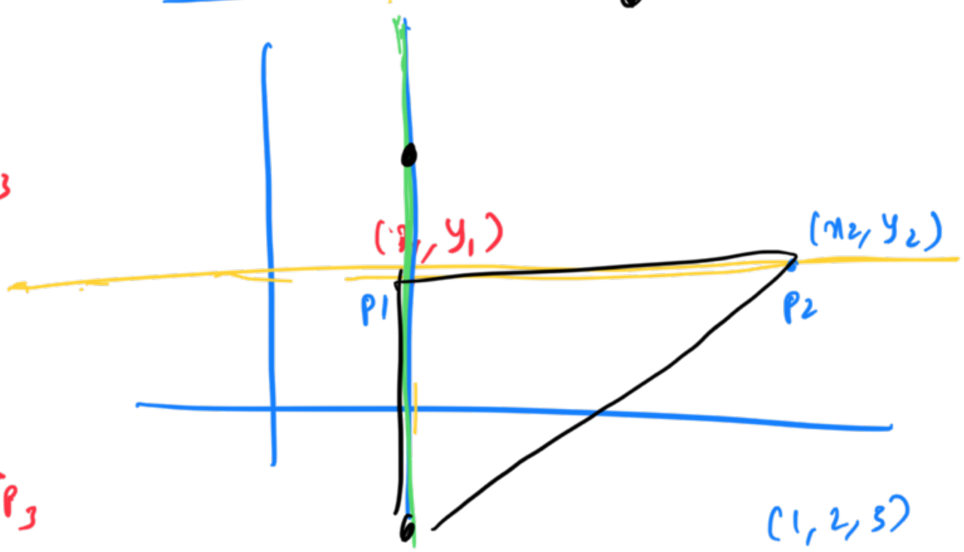
→ (x_2, y_2) to be the point || to y-axis

→ (x_3, y_3) to be the point

```
for(i=0 to n-1){
    for(j=0 to n-1){
        for(k=0 to n-1){
            if(i!=j && j!=k){
                if(x[i]==x[k] &&
                   y[i]==y[j])
                    count++;
            }
        }
    }
}
```

P1 (x_1, y_1) , P2 (x_2, y_2) , P3 (x_3, y_3)

if($x_1 == x_3$ && $y_1 == y_2$)



(1, 2, 3) (3, 2, 1)

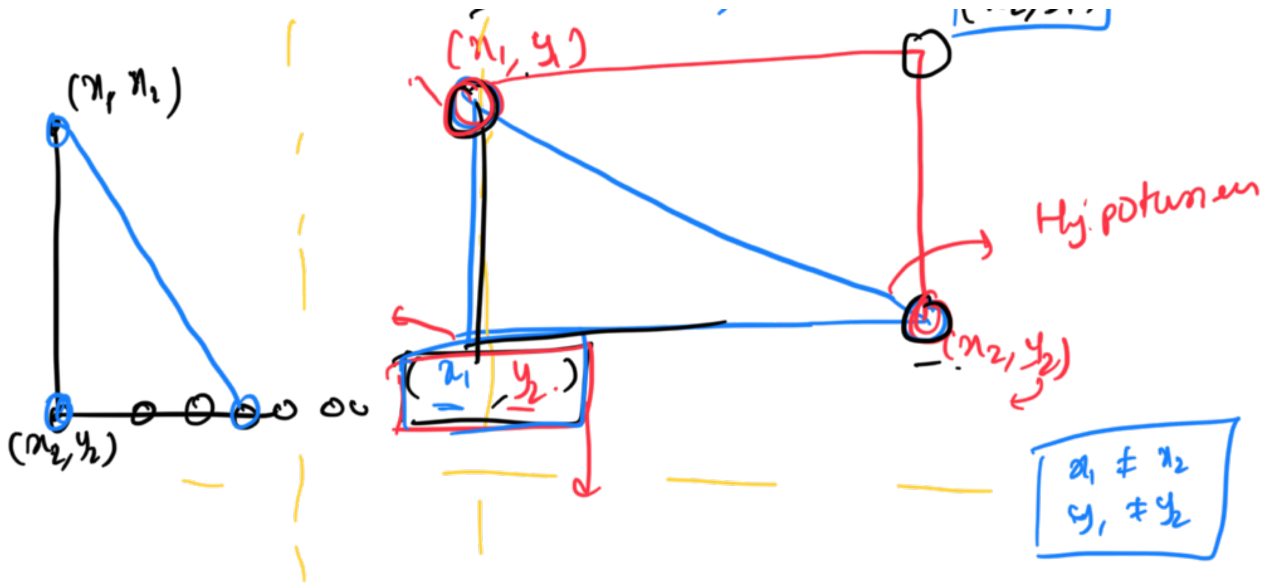
$\boxed{T.C: \; O(n^3)}$

Approach2:

(x_1, y_1) (x_2, y_2) → Part of hypotenuse

Given 2 Points (P_1, P_2) which forms a △

Do we P_3 → (x_3, y_3) $\boxed{(x_2, y_1)}$



for(i = 0 to n-1) {
 for(j = i+1 to n-1) {
  if(x[i] != x[j] && y[i] != y[j]) {
   point1 = [x[i], y[j]]
   if (point1 in hashset)
    count++;
   point2 = [x[j], y[i]]
   if (point2 in hashset)
    count++;
  }
 }
}

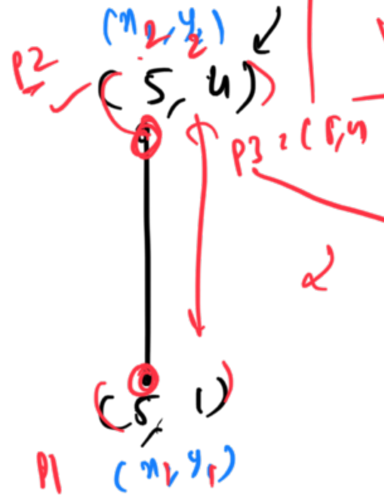
unordered_map < pair<int, int> >

pair < int, int >

T.C: $O(n^2)$

Ordered Map: $O(n^2 \log n)$

Unordered Map: $O(n^2 \times O(1)) = O(n^2)$



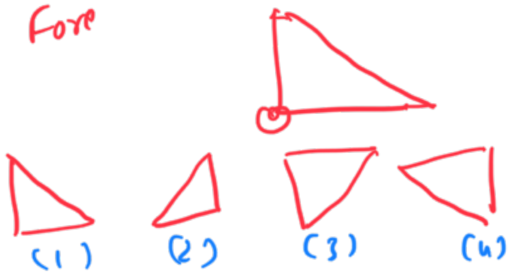
point1 = (x_1, y_2)

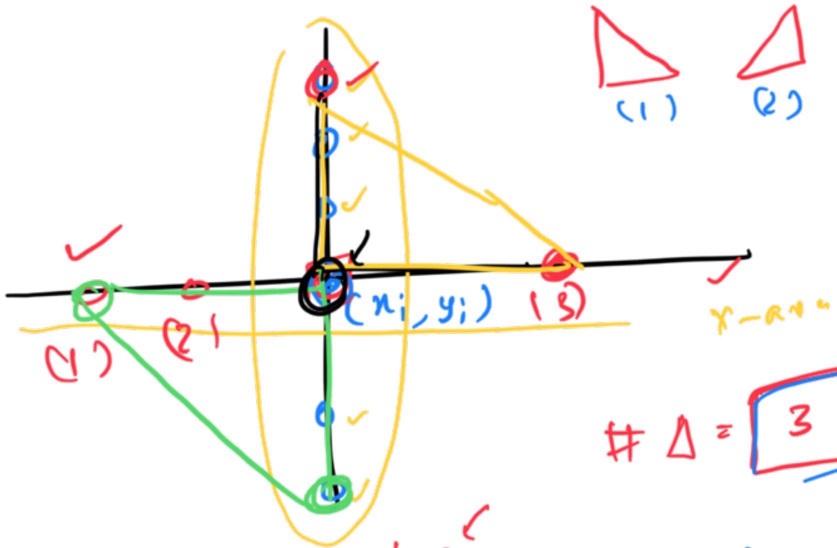
(5, 4)

(True)

Approaches :

Idea is from Brute Force





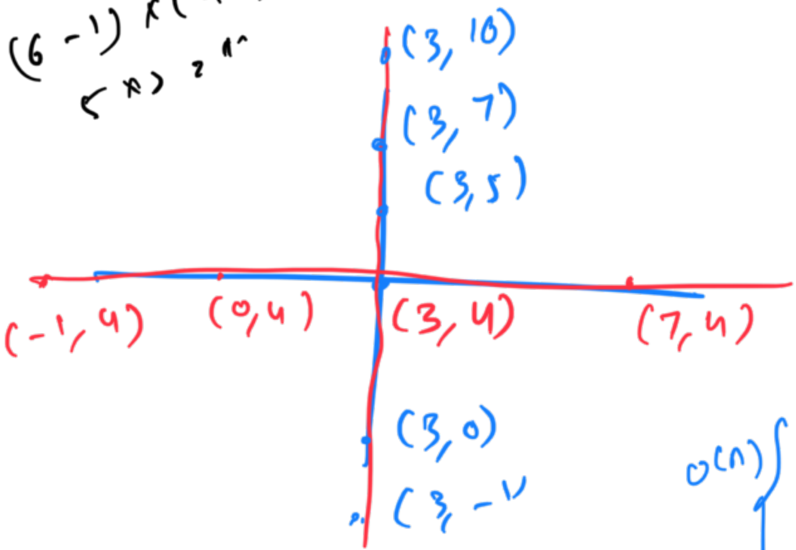
$\# \Delta = \boxed{3 \times 5}$

$\text{freqX} = \{3 : 6\} \longleftrightarrow \text{freqY} = \{4 : 4\}$

6×4

$(6-1) \times (4-1)$

$5 \times 3 = 15$



O(n)
```
for (i = 0; i < n; i++){
    freqX [x[i]] ++;
    freqY [y[i]] ++;
}
```

Key : Integer

O(n)
```
for(i = 0; i < n; i++){
    ans += (freqX[x[i]] - 1) * (freqY[y[i]] - 1)
}
```

$\sum \left(\text{freqX}[x[i]] - 1\right) \times \left(\text{freqY}[y[i]] - 1\right)$

$\forall i$

consider every point to be Hypotenuse Point

T.C : O(n)

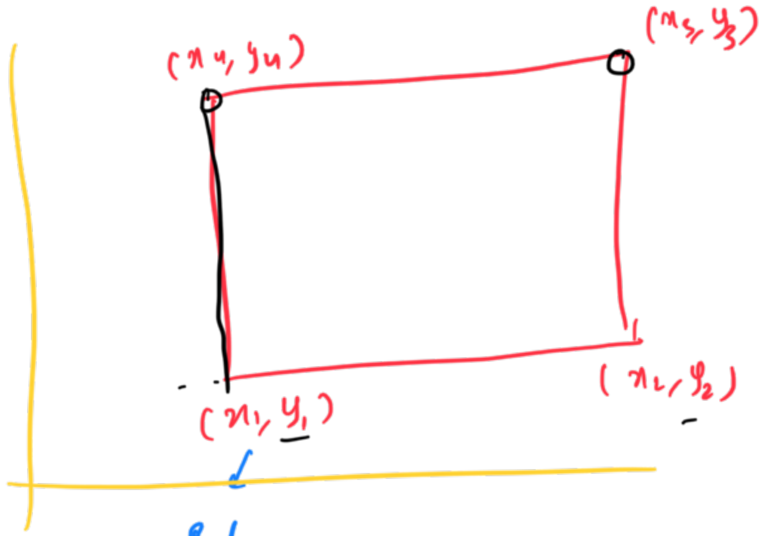
S.C : O(n)

S.C.

Edge Case: $\boxed{(1,2) \quad (1,2)}$

Question: Count no. of Rectangles

Brute Force



$y_3 = y_4$
$y_1 = y_2$.
$x_1 = x_4$
$x_2 = x_3$

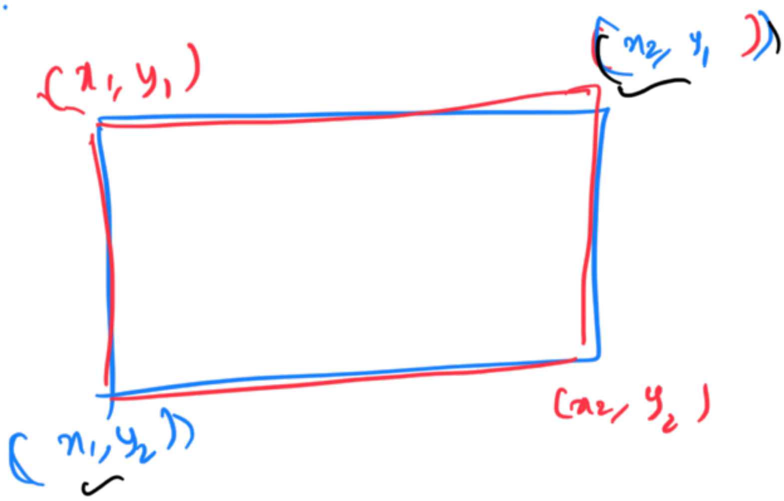
$(x_1, y_1) \to$ BL
$(x_2, y_2) \to$ BR
$(x_3, y_3) \to$ TR
$(x_4, y_4) \to$ TL

$\forall (P_i, P_j, P_k, P_l)$

```
for (i = 0 to n-1)
  for (j = 0 to n-1)
    for (k = 0 to n-1)
      for (l = 0 to n-1)
        if (i != j != k != l):
          // check 4 condition
          count++;
```

T.C: $O(n^4)$

Approach:




```
for (i = 0 to n-1) {
  for (j = i+1 to n-1) {
    if( x[i] != x[j] && y[i] != y[j] ){
      if( (x[i], y[j]) is set && (x[j], y[i]) is in set)
          count++;
    }
  }
}
```

T.C : $O(n^2 \log n)$ (Ordered set)
$O(n^2)$ (Unordered set)

Pattern Matching (strings)

Question: Given 2 string pattern (s) and a text (t) (ctrl - F)

s = 'scaler'
t = 'helloscaler'

How big can the text be?

1) Huge web Page
2) Gmail Inbox

2) ...
3) Plagarism Check

1) check if pattern exists
2) No. of occurrences

pattern = hello
$O(|S|)$

Brute Force

t = "helloscaler bla bla ACB"
scaler

s = "scaler" => 6

$\boxed{T.C: O(|S| \times |t|)}$

length of pattern; length of text

S.C: $O(1)$

< , >

Better Approach:

Easy to compare Numbers / Strings

Numbers: $O(1)$; Strings: $O(len)$

Rabin-Karp:

Map strings to numbers

t = a a b c a d a , $\boxed{S = ada}$

f (cada) = p

f(aab) = P_1

f(abc) = P_2

Requirements of this Mapping

1) Given 2 different strings S_1, S_2

$\boxed{f(s_1) \neq f(s_2)}$

$f(s) = \sum s[i] \times k^{n-i-1}$

Number in base 26

where $\boxed{k \geq 26}$ Size of Alphabet 26

$\boxed{k \geq 52}$

Use Prime

$\boxed{k = 31}$

$N = 4$

$\boxed{k \geq 26 \quad k = 31}$

$s = a\,b\,c\,d$ (97, 98, 99, 100)

$\rightarrow f(s) = 'a' \times k^3 + 'b' \cdot k^2 + 'c' \cdot k + d$

$f(xab) = 'x' \cdot k^2 + 'a' \cdot k + b. \Rightarrow$ Overflow

$\rightarrow$ There are no collisions with this map

to overcome overflow issues,

% M

$\boxed{f(s) \% M} \Rightarrow$ We will have collisions

Polynomial Rolling Hash Function

$h(s) = \sum(s[i] \times k^{n-i-1}) \% M$

$M = 10^9 + 7$

Probability of collisions

$M = 10$

$h(s) = f(s) \% 10$ $\quad (0 \text{ to } M-1)$

$\downarrow$ 0 to 9

$h(s_1) = 3$

0	1	2	3	4	5	6	7	8	9
			s1		s2				

s1 : 0

s2 : $\frac{1}{10}$

s3 : $\frac{2}{10}$

$(n < 10)$

sn : $\frac{n}{10}$

$10^5 - 10^6$ strings

: $Pr(\text{all}) = \boxed{\frac{n}{M}} \Rightarrow \boxed{\frac{10^5}{10^9}} = 0.0001$ $\quad (10^{-4})$

M: 10^9

probability of collision is very very low

$h(s) = f(abc) = (a \times k^2 + b \cdot k + c) \% M$ $\quad O(|s|)$

$O(|s|) \times |t|$

t = ab ab a a a a d f g a b

s = a a d f ✓

$h(aadf) = O(|s|)$ $\quad (a \times k^3 + a \times k^2 + d k + f) \% M$

P = [ ] $\quad$ | k^0 | k^1 | k^2 | k^3 | k^4 | . | . | . | . | $\Rightarrow O(|s|)$

$k^0, k^1, \ldots \ldots k^{|s|-1}$

$\boxed{T.C: \ O(|s| \times |t|)}$

Brute force

for (i=0; i<|s|; i++)

Can we optimise

$t =$ a b c d e f g h i j $s =$ f g h i

'a' × $K^{|s|-1}$

$O(|s|) \leftarrow h(abcd) = ('a' \cdot K^3 + 'b' \cdot K^2 + 'c' \cdot K + 'd') \% M$ — ①

$O(|s|) \leftarrow h(bcde) = ('b' \cdot K^3 + 'c' \cdot K^2 + d \cdot K + e) \% M$ — ②

h1, h2

%M

R.P

$h2 = (h1 - 'a' \cdot K^3) \times K + 'e'$

step1 : $h = (h - (t[0] \times K^{|s|-1} \% M) + M) \% M$

step2 : $h = (h \cdot K) \% M$

step3 : $h = (h + t[4]) \% M$

$\boxed{O(1)}$

sliding window

$O(|s|)$

$t =$ a b c d e f g h i j $s =$ f g h i

if (hash(t) == hash(s)) {
compare each char

}

Average T.C / Worst case T.C

T.C :

{ } t

O(t) +

for (i=0; i< s.length; i++)

O(S) {

Average T.C: $O(|t| + |s|)$

Worst Case T.C :

T.C: $O(|t| * |s|)$

1) Poor Hash function : $O(|t| * |s|)$

$M = 10^9$

t = "aaaaaaaaaaaaaaaaab"

p = "aaa"

$O(|s|)$

worst T.C: $O(|t| * |s|)$

RABIN - KARP

* ?

DP

any n..?

chain

if (Key is not in hashmap){
insert(key)

(2, 1)
(2,2)
(3, 2)

(Key) => Value

Hash Funct

(Integer)

(2, 2)

(3, 2)

hash m

{2, 2}

((2) 3)

(4, 3)

(4, 3)

(2, 3)

Hashmap

Key Val

(2, 3)

23

(4, 3)

(2, 5)

(

26 =>

[0 - 9]

(a, b)

10*a+b

(2 p x 3)

20 + 3

23

(a) (b)

(23) => O(1)

Key

Key : 23

Hashset

1 hashmap

hashmap <int, hashmap>

Expand : O(1)

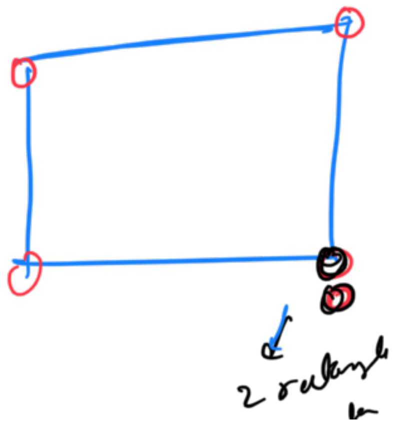
1 hashmap

t = "   " 10^6

p1 = abc : O(t + |p1|)

O(t + |p2|)

p2 = abcd



1)

```
rabinKarp (string s){
      int mod = 1e9 + 9;
      int k = 26;
      int hash_t = 0, hash_s = 0;
      power[|s|] // Stores the powers of k.

      for(int i=0; i< |s| ; i ++){
            hash_t = (hash_t +  pow[|s| - i - 1] *  t[i] ))%mod
            hash_s = (hash_s +  pow[|s| - i - 1] * s[i] ))%mod
      }
      if(hash_t == hash_s && s == t.substr(0, |s|))
            count++;

      for(int i =|s| ; i < |t|  ; i++){
            hash_t =(hash_t - (t[i-|s|] * pow[ |s| - 1])%M + M)%M;
            hash_t = ( hash_t * k ) % M
            hash_t = hash_t +  t[i] )%M

            if(hash_s == hash_t && s == t.substr(i-|s| + 1, |s|)){
                  count++;
            }
      }
      return count;
}
```