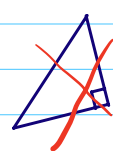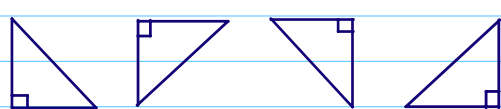3problems: — Count of triangles
— Sum max-min
— window string

Interveiw problems:

P1 Given n points in a 2D plane, count the number of triangles with sides prallel to xy axis. All three corners must be part of input & two triangles are diffirent if one point is different.   points are ditinct
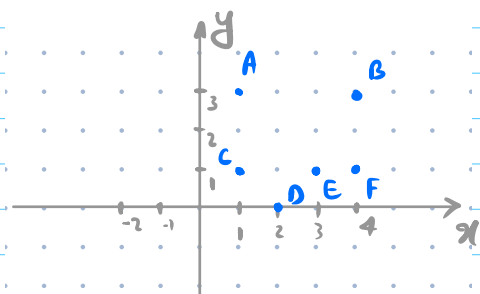

not || to XY axis   right angle

xHM
$[1] \to x\ 2$
$[4] \to x\ 2$
$[2] \to 1$
$[3] \to 1$

0 1 2 3 4 **5**
A B C D E F
$X = \{\ 1\ ,\ 4\ ,\ 1\ ,\ 2\ ,\ 3\ ,\ 4\ \}$
$Y = \{\ 3\ ,\ 3\ ,\ 1\ ,\ 0\ ,\ 1\ ,\ 1\ \}$

yHM
$[0] \to 3$
$[1] \to 1,3,4$
$[3] \to 1,4$

xHM $[1] \to 3,1$ $[4] \to 1,3$
$[2] \to 0$
$[3] \to 1$

ans = 6

$A(x_2, ?)$
$C(?, y_2)$

$\dfrac{x_2\ \ y_2}{B}$

A x_1, y_1
B x_1, y_2 C x_3, y_3
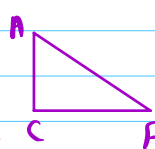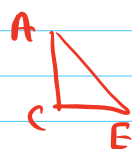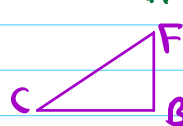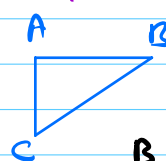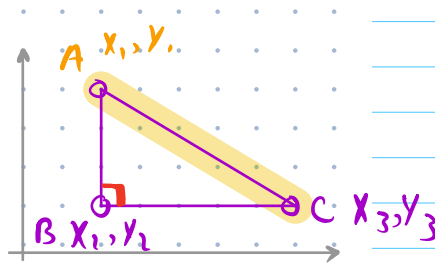
$\begin{cases} X_1 = X_2 \\ Y_2 = Y_3 \end{cases}$ && $X_1\ != X_3$

$i = 0 \to n-1$
$j = i+1 \to n-1$

idea 1   similar to prev solution with modification for ◺

TC: $O(n^2)$
SC: $O(n)$

idea 2

hash map for Freq
x of all point

x HM ←

y HM ←

hash map for Freq
y of all point

$(int) \rightarrow list \quad (int, int, int)$
$\quad \downarrow \quad \quad \quad \searrow O(1)$
$\quad x$

4-1

3-1

$3 \times 2 = 6$
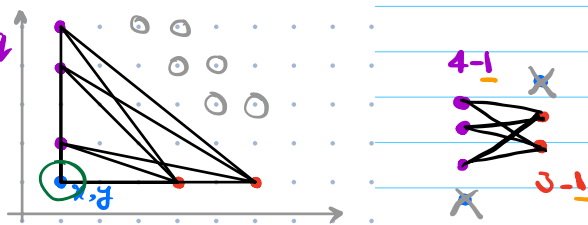
for $i = 0 \rightarrow n-1$
$\quad X[i]$
$\quad Y[i]$

x HM → Hash map for all X coordinations $O(n)$
y HM → Hash map for all Y coordinations $O(n)$

for $i = 0 \rightarrow n-1$ // focus on each corner
$\quad$ ans += $(x HM[X[i]]-1) * (y HM[Y[i]]-1)$
$\quad \{ x HM[X[i]]++$
$\quad \{ y HM[Y[i]]++$

TC:
$O(n)$

exclude the current point i

SC:
$O(n)$

ret ans;

$$5 \begin{vmatrix} max & 4 \\ min & 1 \end{vmatrix} \quad 2 \begin{vmatrix} max & 1 \\ min & 4 \end{vmatrix} \quad 3 \begin{vmatrix} max & 2 \\ min & 2 \end{vmatrix} \text{distinct}$$

☐ P2 Find the **sum** of (max – min) for **all subsets** of

Some learners
say
Quiz not
showing up

Quiz $a: \{2, 3, 5\}$ (indices 0, 1, 2)

$\to 3$ (i)

n element
2^n subset

$2^3 = 8$

brute
force   TC: $O(2^n \times n)$

$2 \leftarrow +\#2 \quad -\#2$
$\qquad\qquad\text{max} \quad\quad \text{min}$

$$\sum_i +\#a[i] - \#a[i]$$
$\qquad\quad \text{max} \qquad \text{min} \leftarrow$ Contribution technique
$O(1)$

	(+) max	(−) min	max_min
{ }	x	x	x
{3}	3	3	0
{2}	2	2	0
{5}	5	5	0
{3,2}	3	2	1
{3,5}	5	3	2
{2,5}	5	2	3
{3,2,5}	5	2	3

(each row marked +)

$9 \leftarrow$ ans

#of
subset
6 is max

$\{3, 2, 8, 7, 4, 6\} \rightarrow \{2, 3, 4, 6, 7, 8\}$ (indices 0 1 2 ③ 4 5)
2^i Quiz

#of
subset
6 is min

$2^{(n-1-i)}$ 2^2

$\{3 \; 2 \; 8 \; 7 \; 4 \; 6\} \rightarrow \{2, 3, 4, 6, 7, 8\}$ (indices 0 1 2 3 4 5)
Quiz   4

$\{6, 7, 8\}$
$\{6\}$
$\{6, 7\}$
$\{6, 8\}$

TC: $O(n + n\log n) \sim O(n\log n)$

for loop
over
i

① Sort
② $\sum_i +2^i \times e - 2^{n-i-1} \times e = $ ans
at $\;\;e$
index $\;i$

② $\boxed{\text{ans} += 2^i \times a[i] - 2^{n-i-1} \times a[i]}$

size n   size m $n \leq m$

P3 Given 2 strings A and B, find the length of smallest substring in B which contain all characters of A. Return -1 if not possible

→ order desn't matter

ex A = "a b"
B = "A a 9 1 b" → ans = 4

→ order desn't matter

ex A = "a b"
B = "A b 9 1 a" → ans = 4

ex A = "b b"   Count matters
B = "A b 9 1 b" → ans = 4

ex A = "a b c"
B = "A b 9 1 b"   ans = -1

ex A = "a b c"
B = "a d o b e c o d e b a n c"   ans = 4
        6   4

idea A "a b c c" | a → 1   ans = 8 ̸7 6   a → 1 → Afreq
                     | b → 1 → Afreq   b → 1
                     | c → 2   "a b c c"   c → 2

B freq
↑ 1
| a → 0 b → 1   B "a c x y b c y h l"a   "s r a c x y b c y h l"
| c → ̸1 2   l   a   l   r
| x → 1   g → 1 l → 1   r
| y → 1   h → 1   → window that is minimal answer   ̸x ̸1
WFreq   ∀ key in A Freq[key] ≤ WindowFreq[key]   ̸x ̸1
                                                                  ̸a ̸1
a, 1   (a, 1)   A "a b c c"   c → ̸1 2
c, 2   (b, 1)   x → 1
x, 1   (c, 1)   y → 1
y, 1   B "a c x y b c y h l a b c c"   b → 1
y, 1
b, 1

ans = INT_MAX

psudo code

```
ans = INT_MAX

for i = 0 → n-1
    Afreq [ a[i] ]++;
```
→ start a **variable window** from index 0

```
Bfreq [ b[0] ]++
l = r = 0
while ( r < m && l < m ){
```
→ end of string B

```
    if ( hashMapLessOrEqual ( Afreq , Bfreq ) ){  // O(n)
```
∀ key in AFreq[key] <= WindowFreq[key]

reduction {
```
        ans = Min ( ans, r-l+1 )
        Bfreq [ b[l] ] -= 1
        l += 1
    } else {  // I still don't have enough chars in
                              cur. window
```

expansion {
```
        r += 1
        if ( r == m ) break
        Bfreq [ b[r] ] += 1
    }
}
```

ex.

A ↘
'c'

B ↘
'ac'
 r   l

A   abc

B   xyabacy bca
    0 1 2 3 4 5 6 7 8 9
    l   r

TC: O(m x n)
SC: O(m + n)

$$a : \{ \overset{0}{2}, \overset{1}{3}, \overset{2}{5} \}$$

for loop
over
i

at
index $\to$ i

e
i

① Sort

② $\sum_i + 2 \times e^i - 2^{n-i-1} \times e = ans$

② $ans += 2^i \times a[i] - 2^{n-i-1} \times a[i]$

i	a[i]	ans
0	2	$+= 2^0 \times 2 - 2^2 \times 2 = -6$
1	3	$+= 2^1 \times 3 - 2^1 \times 3 = 0$
2	5	$+= 2^2 \times 5 - 2^0 \times 5 = 15$

9