Q.1

A[ ] = { ✓ ✗ . . . . }

← n →

Generate all possible subsets/subseq.

A[ ] = { ↙ ↘ ↙ ↘ . . . . }

✓ ✗ ✓ ✗

2 · 2 · 2 · 2 . . . 2 ⇒ 2^N

$$A[\,] = \{ \overset{0}{1}, \overset{1}{2}, \overset{2}{3} \} \Rightarrow \{ \overset{0}{3}\ \overset{1}{2}\ \overset{2}{1} \}$$

$$A[\,] = \{ \overset{2}{3}, \overset{1}{2}, \overset{0}{1} \}$$

N=3

$2^N = 2^3 = 8$

- generate 2^N nos

0 to $2^N - 1$

0 to $2^3 - 1$

i.e 0 to 7

0	< 0 0 0 >	{ }
1	< 0 0 1 >	{1}
2	< 0 1 0 >	{2}
3	< 0 1 1 >	{1, 2}
4	< 1 0 0 >	{3}
5	< 1 0 1 >	[{1, 3}]
6	< 1 1 0 >	{2, 3}
7	< 1 1 1 >	{1, 2, 3}

2nd 1st 0th

5 → [1 0 1]

{ A[0], A[2] }

{ 1, 3 }

N=3

$$\begin{array}{cccc} \text{idx} & 0 & 1 & 2 \\ A[\,] = & \{1 & 2 & 3\} \end{array}$$

0 to 7

2nd 1st 0th

5 < 1 0 1 >

bitPos = 0

1 → 0 0 1

A[bitPost]

A[0] { 1 }

→ 5 & 1 == 1

(5) 1 0 1

(1) & 0 0 1

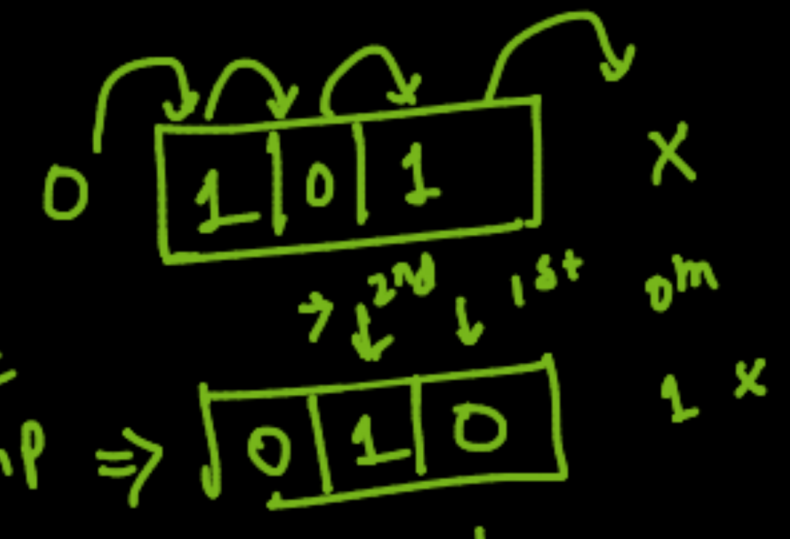
(1) 0 0 1

2nd 1st 0th

5 → 1 [0] 1

bitPos = 1

5 >> 1 ⇒



temp ⇒ 0 1 0

& 0 0 1

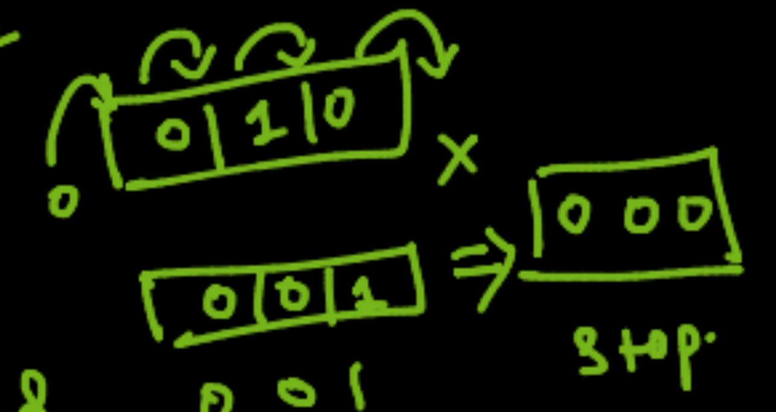
0 0 0 = 0

5 >> 1

& 1

(5 >> 1) >> 1

bitPost = 2



& 0 0 1

0 0 1

A[bitPost]

A[2] = {3}

⇒ { 1, 3 }

$$A[\,]\ \{\overset{0}{1},\ \overset{1}{2},\ \overset{2}{3}\}$$

0 to 7

$$6 \rightarrow (\overset{2^{nd}}{1}\ \overset{1^{st}}{1}\ \overset{0^{th}}{0})$$

bitPos = 0

6 & 1 ⇒ $\boxed{110}$ >>1

001

000

bitPos = 1

$\boxed{0|1|1}$ >>1

& 001

001 = 1

$\{A[1],\ A[2]\} \Rightarrow \{2, 3\}$

bitPos = 2

$\boxed{0|0|1}$ >>1

& 001

001

000 → stop

N = 3 → 0 to $2^3 - 1$

0 to $\underbrace{2^N}_{\uparrow} - 1$

calculate ?

1. # include <cmath>

① pow(a, b) ⇒ a^b

pow(2, 3) ⇒ $\boxed{2^3}$

* 1 << n ⇐ 2^N

0 0 0 1 << 1

0 0 1 0 <<1 ⇒ 2

0 1 0 0 <<1 ⇒ 4(2^2)

1 0 0 0 ⇒ 8(2^3)

1 << n ⇒ 2^n

< 6 bitwise operators >

operand1 op operand2

binary

- & → AND → 1 when both bits are 1, otherwise it is 0
- | → OR → 0 when both bits are 0, otherwise it is 1
- ^ → XOR → 1 when bits differ, otherwise it is 0

unary

- ~ → flips / NOT → ~1 = 0; ~0 = 1

binary operators

- >> → right shift $x >> p \Rightarrow x / 2^p$ → int. division
- << → left shift $x << p \Rightarrow x \cdot 2^p$

AND, OR, XOR → commutative & associative

$x \& y = y \& x$

$x | y = y | x$

$x \wedge y = y \wedge x$

$x \& (y \& z) = (x \& y) \& z$

$x = 5$

$$\begin{array}{r} 101 \\ \wedge\ 000 \\ \hline 5 \rightarrow 101 \\ \hline \end{array}$$

- * $x \& x = x$
- * $x | x = x$
- * $x \wedge x = 0$

$x = 5$

$$\begin{array}{r} 101 \\ \wedge\ 101 \\ \hline 000 \\ \hline \end{array}$$

$x \& 0 = 0$

$x | 0 = x$

$x \wedge 0 = x$

$N=5$

$$A = [\overset{0}{1}\ \ \overset{1}{2}\ \ \overset{2}{3}\ \ \overset{3}{4}\ \ \overset{4}{5}]$$

$$\underset{\Downarrow}{{}^nC_2} + \underset{\Downarrow}{{}^nC_1} \Rightarrow \frac{n(n+1)}{2}$$

$$\frac{n(n-1)}{2} + n \nearrow$$

$$A[\,] = [\overset{0}{1} \rightarrow \overset{1}{②}\ \ \overset{2}{\underline{\underline{3}}}\ \ \overset{3}{\underline{\underline{4}}}\ \ \overset{4}{\underline{\underline{5}}}]$$

$\text{sum} = \cancel{0}\ \cancel{1}\ \cancel{3}\ \cancel{6}\ \cancel{10}\ 15 \Rightarrow O(1)$

$$\frac{n(n+1)}{2} \cdot 1 \Rightarrow O(n^2)$$

$$A[\,] = [\overset{0}{1}\ \ \overset{1}{2}\ \ \overset{2}{3}\ \ \overset{3}{4}\ \ \overset{4}{5}]$$

$\Rightarrow$ 1. $(i, j) \rightarrow$ iterate over the array from ith index to jth index and add the elements

$i = 0 \qquad j = n-1$

$i = 0 \qquad j = 4 \Rightarrow \underline{\underline{n}}$ steps (worst case)

$$\overset{\sim n^2}{\overset{\Uparrow}{\frac{n(n+1)}{2}}} \cdot n \sim \underline{\underline{O(n^3)}}$$

<u>Limitation</u> $(i, j) \Rightarrow$

$$A[\,] = [\overset{0}{1}\ \ \overset{1}{2}\ \ \overset{2}{3}\ \ \overset{3}{4}\ \ \overset{4}{5}]$$

$$\hookrightarrow (\underset{i}{\overset{\uparrow}{0}}, \underset{j}{\overset{\uparrow}{4}}) = \Rightarrow \boxed{S_{03}} + \underline{\underline{A_4}}$$

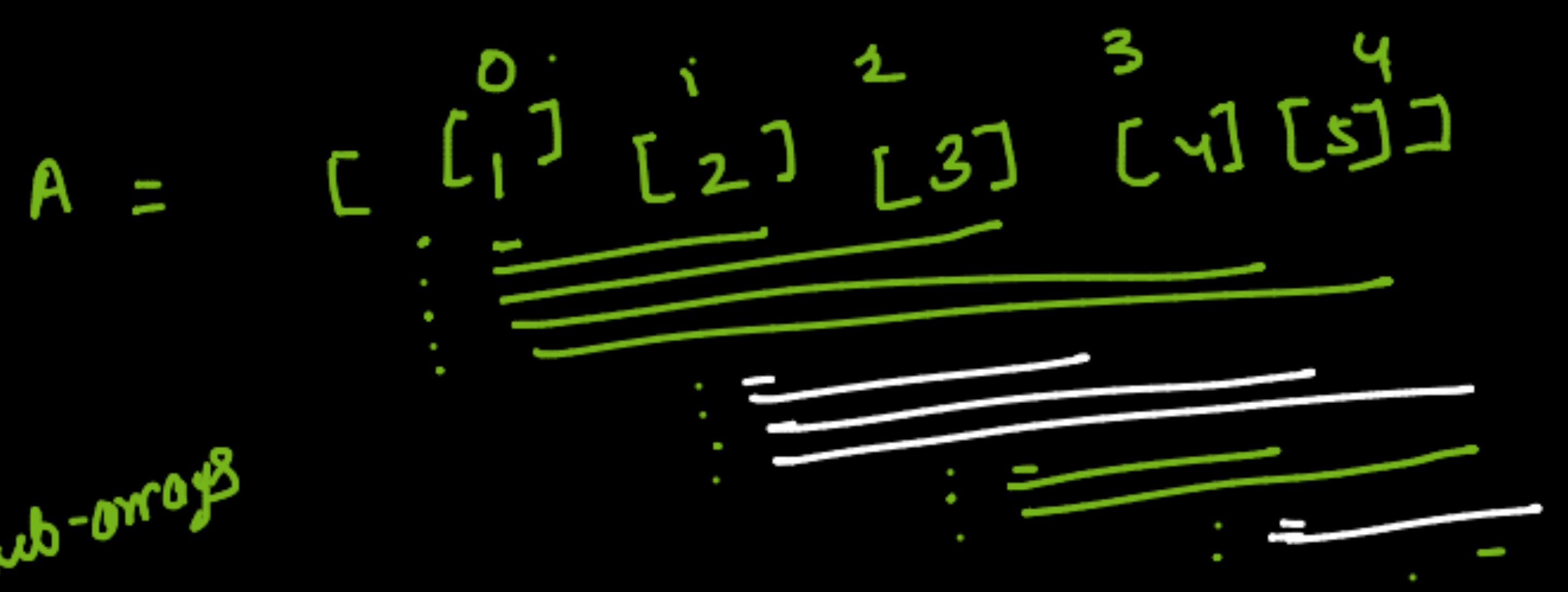
Not reqd'

$$S_{03} = \boxed{S_{02}} + A_3$$

$$S_{02} = \boxed{S_{01}} + A_2$$

$$S_{01} = \boxed{S_0} + A_1$$

$$S_0 = A_0$$

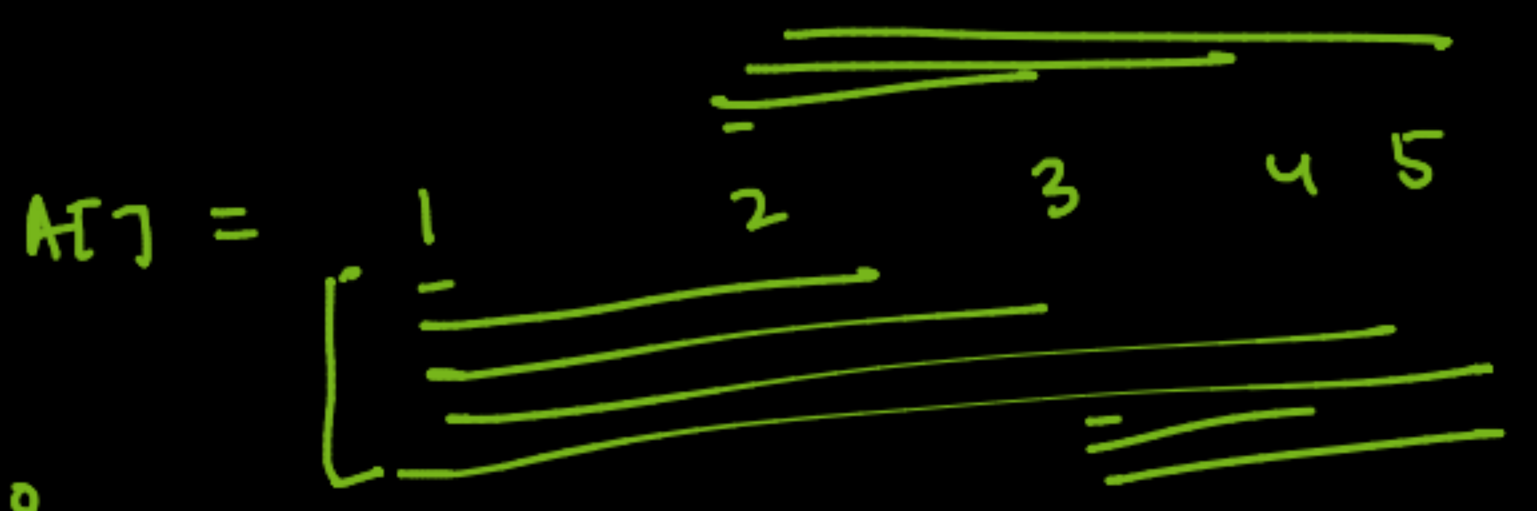


N=5 ⇒ 15 sub-arrays

$${}^{N}C_2 + {}^{N}C_1$$

$$= \frac{N(N-1)}{2} + N$$

$$= \frac{N(N+1)}{2} \Rightarrow \quad \frac{5(5+1)}{2} = 15$$



sum=0

(X)

sum=0

X

(Y)

Sub-array → continuous + order is maintained

Sub-seq / subset → order is mained + may / may not be continue

A[ ] = {1, 2, 3, 4, 5}

{1, 2, 3} ✓ sub.

{1, 3} X. ✓ sub.

(i, j)

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [1 & 2 & 3 & 4 & 5] \\ \uparrow & & & & \uparrow \\ i & & & & j \end{array}$$

pre-compute this

store somewhere

$S_{04} = \boxed{S_{03}} + A_4$

$S_{03} = S_{02} + A_3$

$S_{02} = S_{01} + A_2$

$S_{01} = S_0 + A_1$

$S_0 = A_0$

$$A[\,] = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [1 & 2 & 3 & 4 & 5] \end{array}$$

$$csum[\,] = \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ [0 & 1 & 3 & 6 & 10 & 15] \end{array}$$

pre compute

$(i, j) \rightarrow \underbrace{csum[j+1] - csum[i]}$

$i=1 \quad j=3 \rightarrow csum[4] - csum[1]$

$= 10 - 1$

$= 9$

$$\begin{array}{lcccccc} idx & 0 & 1 & 2 & 3 & 4 & \\ A[\,] = & [1 & 2 & 3 & 4 & 5] & \Rightarrow n \\ csum[\,] = & [0 & 1 & 3 & 6 & 10 & 15] \Rightarrow n+1 \end{array}$$

prefix

csum[i] ⇒ ? sum of elements upto me in index excl.

$$\left[\; csum[i] = csum[i-1] + A[i-1] \;;\; i > 0 \;\right]$$

$$A[\,] = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [1 & 2 & 3 & 4 & 5] \end{array}$$

$$csum[\,] = \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ [0 & 1 & 3 & 6 & 10 & 15] \end{array}$$

$(i=1 \quad j=3) \Rightarrow csum[4] - csum[1]$

$A[\,] = [\qquad\qquad]$

$\longleftarrow n \longrightarrow$

$\#$ good sub-arrays

(i, j)

$$\boxed{S_{ij} \% N = 0}$$

$$S_{ij} = \underline{\underline{c}}sum[j+1] - csum[i]$$

$\underline{\underline{N^2}} \nearrow 2nd$, $\searrow \underline{\underline{3}}rd$ $\rightarrow$ Ⓝ steps.

$(\underline{\underline{i, j}}) \rightarrow \underline{\underline{S_{ij}}} \% \underline{\underline{n}} = 0$

$\Rightarrow \underline{(csum[j+1] - csum[i])} \% n = 0$

$\Rightarrow \underline{\underline{csum[j+1] \% n = \underline{csum[i] \% n}}}$

$\Downarrow$

(i, j) is a good sub array

$\boxed{N=5}$

0 1 2 3 4

A[ ] = [1 [[2 3] [6 4]]

0 1 2 3 4 5

csum[ ] = [0 1 3 $\boxed{6}$ = 12 16]

0 $\boxed{1}$ 2 $\boxed{3}$ 4 5

csum % n[ ] = [0 ① 3 ① 2 ①]

i j+1 i j+1

i = 1 j+1 = 5 ⇒ j = 4

i = 3 j+1 = 5 ⇒ j = 4

⇒ [csum[j+1] % n = csum[i] % n]

⇒ (i, j) is a good subarray

i = 1 j+1 = 3 ⇒ j = 2

⇒ $^3C_2$ ⇒ 3 good subarrays

csum % n [ ] = [y x y x y x y z z]

total no. of good subarrays = $^3C_2$ + $^4C_2$ + $^2C_2$

x > 1 ⇒ $^xC_2$

csum % n [ ] = [0 1 3 1 2 1]

✓0 → 1 ✓3 → 1

1 → ③ → $^3C_2$ ✓2 → 1

⇒ $^3C_2 = 3$

⇒ csum[ ] % n = [0 1 3 7 2 1]

0 to n-1

when you do % n , you can get n unique val from 0 to n-1

=5 freq-array (n) [~~0~~1 | ~~0~~~~1~~ 2 3 | ~~0~~1 | 1~~0~~ | 0]

0 1 2 3 4

freq = [1 | 3 | 1 | 1 | 0]

0 1 2 3 4

cnt = 0

freq-array! = [1 | ③ | 1 | 1 | 0]

0 1 2 3 4

cnt += $^3C_2$

= $^3C_2$

= $\boxed{3}$ ⇐ final ans

idx 0 1 2 …… n-1

A[] = [ ] ⇒ n elements

idx 0 1 2 …… n

csum[] = [ ] ⇒ n+1 elements

idx 0 1 2 …… n

csum%n[] = [ ] ⇒ n+1 elements

0 1 2 … n-1

freq[] = [ ] ⇒ n elements

$\underline{1} = x$

n>1 ⇒ cnt += $^{n}C_2$

Pigeon Hole Principle

N Holes

↑

N+1 Pigeons

N=4 N+1=5

N= 3 1+N = 4

tell me!

⇒ what will be minimum # of good sub arrays?

⇒ csum[]%n ⇒ n+1 ⇐ [… x ….] 0 to n

⇒ freq[] ⇒ n ⇐ freq[x]++

There will be atleast 1 hole marked > 1 value