

- 8 weeks, 16 sessions

Language Problem solving.  
C++ , Java.

→ Class notes, github link.

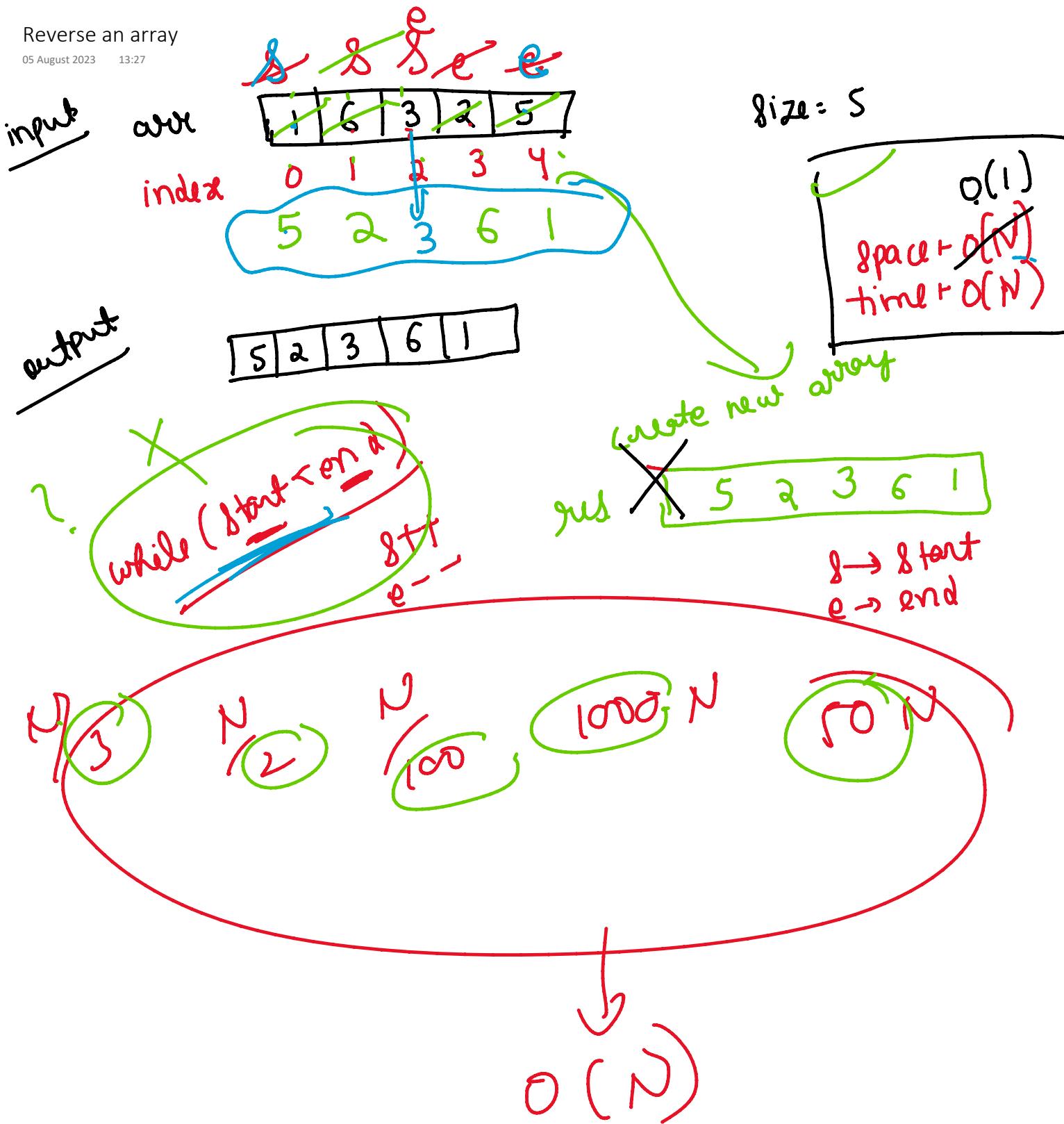
### Syllabus

- 1 → Arrays
- 2 → Arrays + Matrix
- 3 → Searching, Sorting
- 4 → Strings
- 5 → Revision
- 6 → Backtracking
- 7 → Time / Space Comp + Hashmap
- 8 → linkedlist +

- 9 → linkedlist + bit
- 10 → Stacks & Queues
- 11 → Binary trees
- 12 → Trees + BST
- 13 → Heap + Graphs
- 14 → Graphs
- 15 → Greedy + DP
- 16 → DP

## Reverse an array

05 August 2023 13:27



## Missing number

05 August 2023 13:40

$n$  values, range  $[0, n]$   
exactly one value will be missing.

array = [3, 1, 0, 4]  
 ✓ ✓ ✓ ✓

Output = 2

array = [0, 3, 2, 1]

Output = 4

array sum = 8

sum of  $n$  natural numbers =  $n \times (n+1)/2$   
 actual

$$= 4 \times (5)/2$$

$$= 10$$

actual sum - array sum = 10 - 8  
 = 2

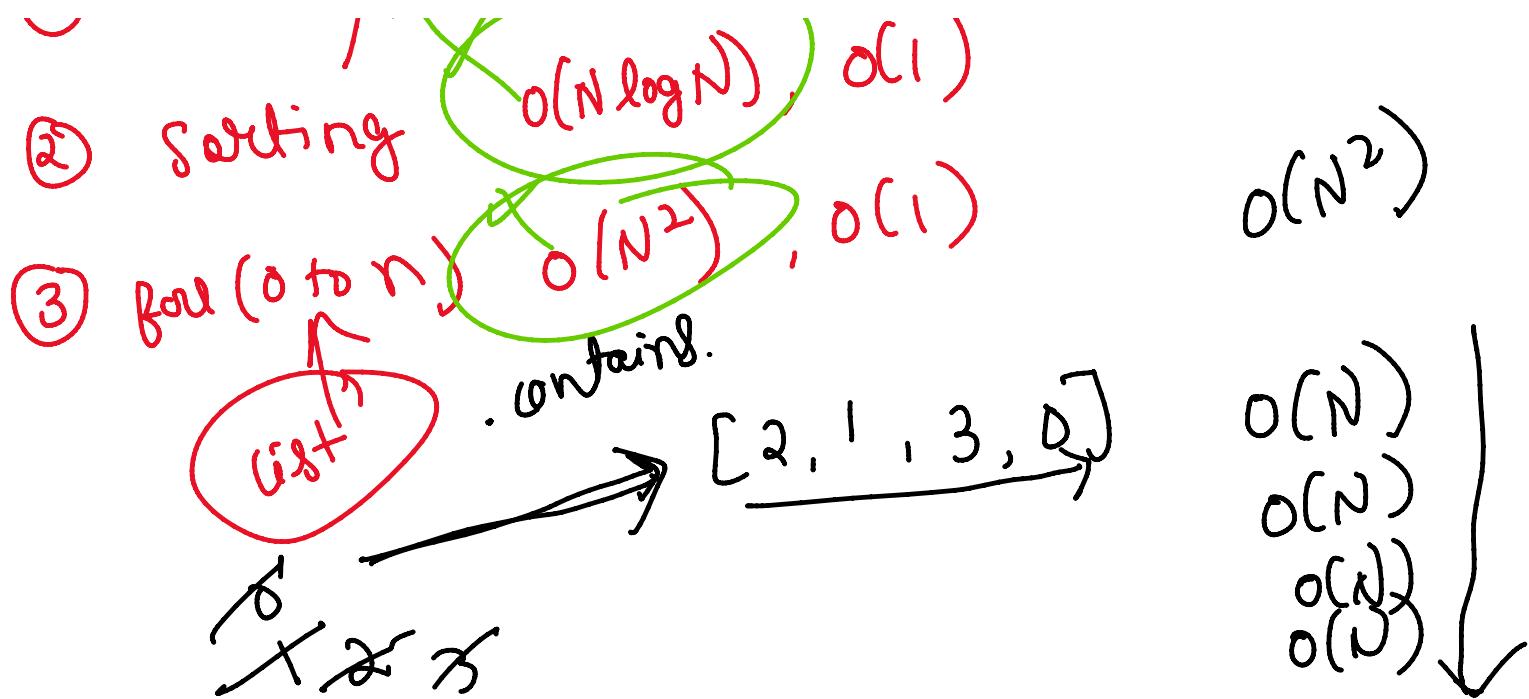
Time  $O(N)$   
 Space  $O(1)$

array = [4,

Time  $O(N)$ , space  $O(N)$

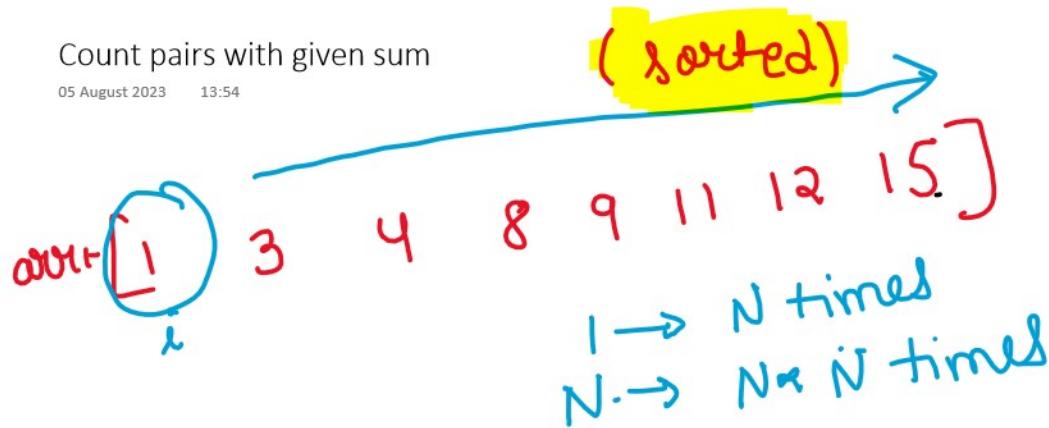
① Hashmap / Array  $O(N)$ ,  $O(N)$   
 $\dots$   $O(N \log N)$ ,  $O(1)$

0 1 2 3 4 5



Count pairs with given sum

05 August 2023 13:54



unique values

$$\text{target} = 13$$

[12, 1]

[9, 4]

ans = 2

(-1)

1. Two for loops.

for ( $i=0$  to  $i=n-2$ )  
for ( $j=i+1$  to  $j=n-1$ )  
 $\text{arr}[i] + \text{arr}[j] ==$

Time  $\leftarrow O(N^2)$   
Space  $\leftarrow O(1)$

$O(N^2)$

2. Hash map.

Time  $\leftarrow O(N)$   
Space  $\leftarrow O(N)$

$O(1)$

3. Two pointer.

Time  $\leftarrow O(N)$   
Space  $\leftarrow O(1)$

sort → Two pointer  
→ Binary Search.

$$\text{target} = 13$$

[1, 3, 4, 8, 9, 11, 12, 15 ]

$$\text{currSum} = 16 \cancel{13} \cancel{14} \cancel{12} 13$$

8++  
e--

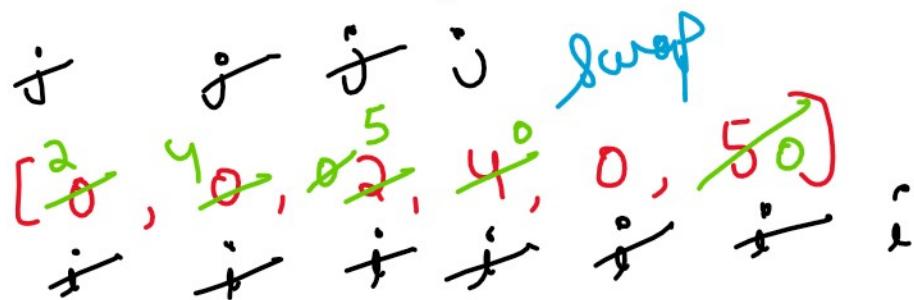
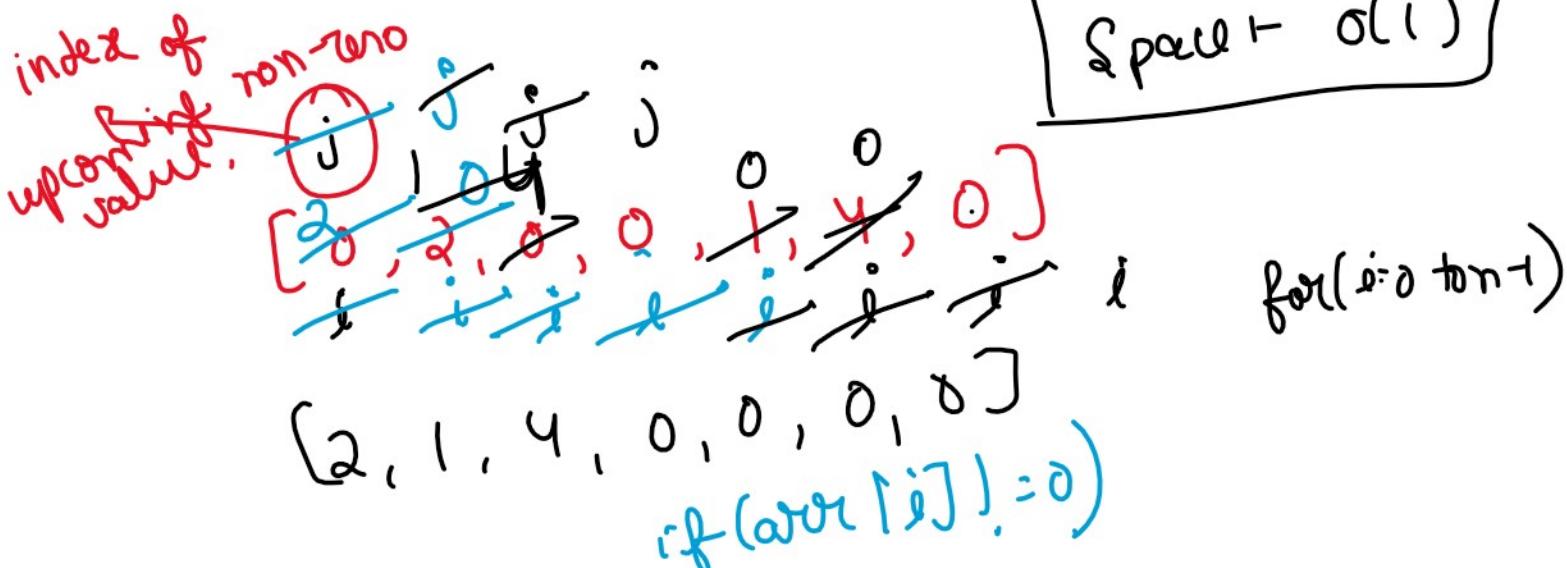
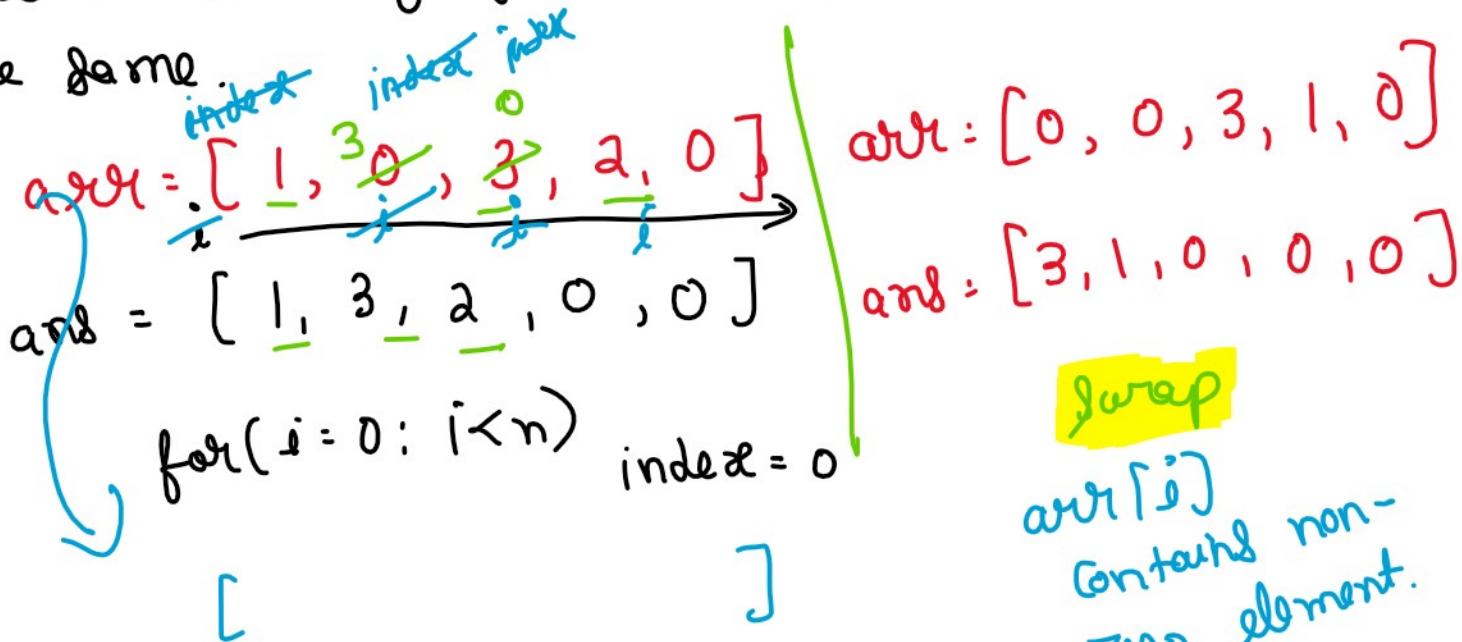
~~Count =  $\sigma' / 2$~~

L<sup>c</sup>  
while (start < end)

## Move zeros

05 August 2023 14:16

move zeros towards the end of the array.  
relative ordering of remaining elements should  
be same.

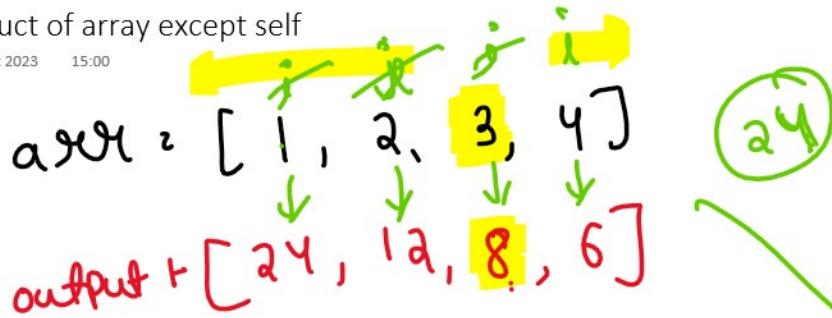


[2, 4, 5, 0, 0, 0]

[~~1, 5, 4, 3~~]

## Product of array except self

05 August 2023 15:00



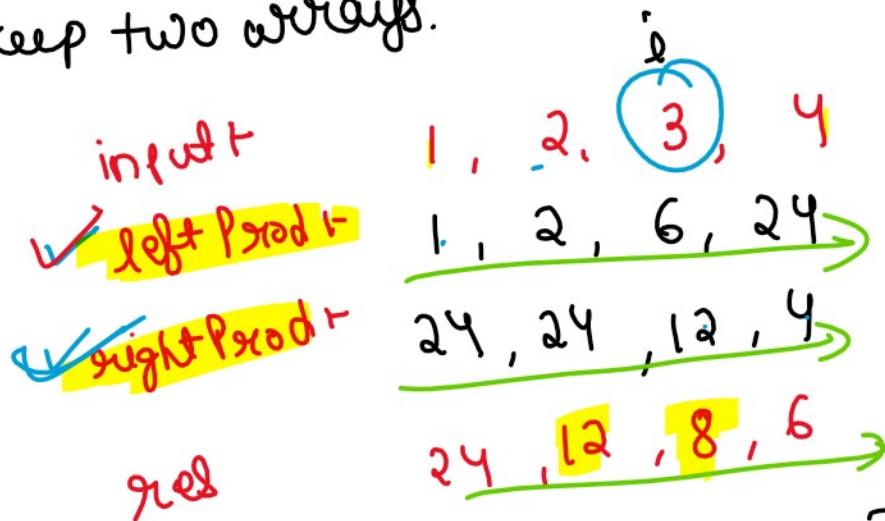
we cannot use division operation  
ignore space due to output array.

$$\text{array} = [-1, 1, 0, -3, 3]$$

$$\text{output} = [0, 0, 9, 0, 0]$$

$O(N)$   
 ~~$O(N^2), O(1)$~~

1. Two for loops. Both from 0 to  $n-1$
2. keep two arrays.



~~$O(1)$~~   
Space  $\rightarrow O(N)$   
Time  $\rightarrow O(N)$

$$res[i] = \frac{\text{leftProd}[i-1]}{1 \times 12}$$

$$P + N + N = 3N$$

$$= O(N)$$

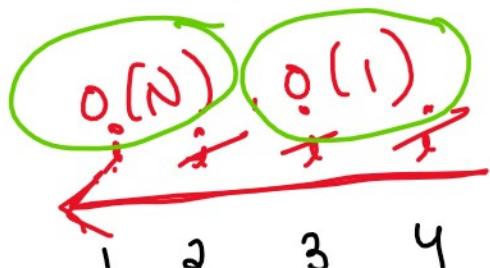
$$\times \cancel{\text{rightProd}[i+1]}$$

$$N + N$$

$$\Rightarrow 2N$$

$$\Rightarrow O(N)$$

3. Optimised.



input:  $1, 2, 3, 4$   
 res:  ~~$1, 2, 6, 24, 6$~~   
 ~~$24, 12, 8, 24$~~

$\text{rightProduct} = 1 \times 4 \times 2 \times 24$   
 $\text{res}[i] = \text{res}[i-1] \times \text{rightProduct}$   
 $\text{res}[3] = 6 \times 1 = 6$   
 $\text{res}[2] = 6 \times 4 = 8$   
 $\text{res}[1] = 1 \times 12$

$i=1$   
 $\text{arr} = \{ 3, 2, 5, 1 \}$   
 $\rightarrow \text{res} = \{ \cancel{3}, \cancel{6}, \cancel{30}, \cancel{30} \}$  10, 15, 6, 30

$\text{rightProduct} = 1 \times 1 = 1 \times 5 = 5 \times 2 = 10$   
 $\text{res}[i] = \text{res}[i-1] \times \text{rightProduct}$   
 $\text{res}[3] = 30 \times 1 = 30$   
 $\text{res}[2] = 6 \times 1 = 6$   
 $\text{res}[1] = 3 \times 5 = 15$

$[1, 2, 3, 4]$

**subarray**: continuous part of array.

$(2, 3)$ ,  $[2, 3, 4]$ ,  $\cancel{[1, 4]}$ ,  $[1]$ ,  $[1, 2, 3, 4]$

**subsequence** need not be continuous. we can skip elements.  
But ordering should be same.

$[2, 4]$ ,  $[1, 3]$ ,  $[1, 2, 3]$ ,  $[4]$ ,  $\cancel{[3, 2]}$

**subset** = subsequence + empty set

$\{2, 4\}$ ,  $\{1, 3\}$ ,  $\{1\}$ ,  $\{\}$

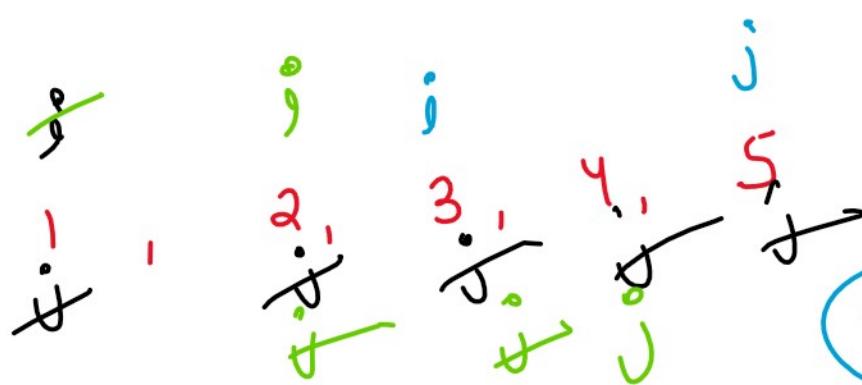
$\boxed{1, 2, 3, 4}$

generate all possible subarrays.

1	2	3	4
$1, 2$	$2, 3$	$3, 4$	
$1, 2, 3$	$2, 3, 4$		
$1, 2, 3, 4$			

2 loops.

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



$O(N^3)$   
 $O(1)$

$3, 4, 5$

1,

1, 2

1, 2, 3  
1, 2, 1, 3, 4  
1, 2, 1, 2, 3, 4, 5

2  
2, 3  
2, 3, 4

2, 3, 4, 5

Starting

ending

point

for ( $i=0$  to  $n-1$ )

for ( $j=i$  to  $n-1$ )

for ( $k=j$  to  $j$ )