

"To improve is to change,  
to perfect, is to change often!"

- "House of cards"

## \* Problem constraints!

Q.  $T.C = O(\log N) \Rightarrow$  (binary search)



Can not ask T.C. constraint during the interview.

Q1  $\underline{N \leq 10^6} \Rightarrow O(N^2) \rightarrow \text{TLF}$   
 $\Rightarrow O(N), O(N \log N)$

coding platform

Q2  $N \leq 20 \Rightarrow O(N) \checkmark$   
 $\Rightarrow O(N^2) \checkmark$   
 $\Rightarrow O(N^3) \checkmark$   
 $\Rightarrow O(2^N) \times$   
 $\Rightarrow O(!N) \times$

$10^8 - 10^9$  iteration

$$\begin{aligned} 2^{20} &\rightarrow 2^{10} \Rightarrow 10^3 \\ (2^{10})^2 &\Rightarrow (10^3)^2 \\ 120 > 10^9 &\Rightarrow 10^9 \end{aligned}$$

Q3.  $N \leq 10^{10} \Rightarrow O(N) \times$  (would give TLE)  
 $\Rightarrow O(\log N), O(\sqrt{N}), \underline{O(1)}$

Q. Given an array of 1's and 0's. we can replace one of 0 with a 1. Return the count of max consecutive 1's in the array.

Amazon  
Microsoft

Adobe  
Dhiraj

ex:

$$A = [1, 1, 0, 1, 1, 0, 1, 1, 1]$$

(I)

$$\Rightarrow [1, 1, 1, 1, 1, 0, 1, 1, 1]$$

5

(II)

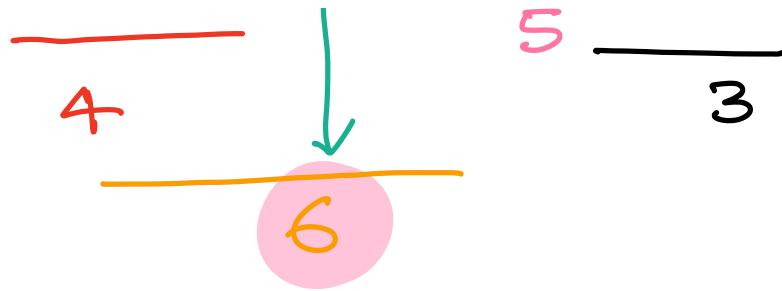
$$[1, 1, 0, 1, 1, 1, 1, 1, 1]$$

6

Quiz:

$$A = [0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0] \quad ans = 6$$

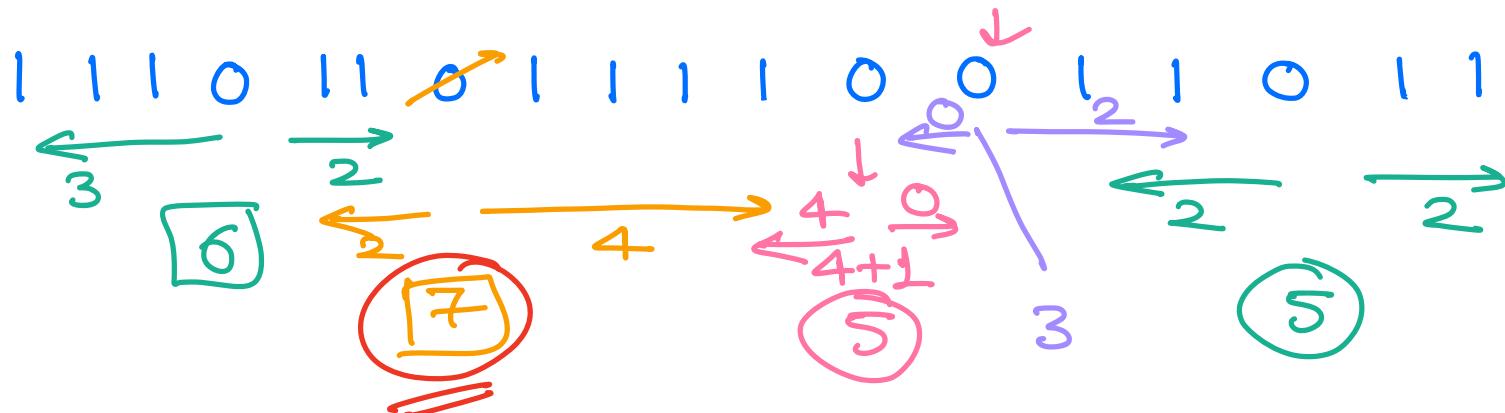
↓ | \_\_\_\_\_ ↓ ↓



Brute force idea:

- ① when we see a zero. left-count
- ② count how many 1's are on the left
- ③ count how many 1's are on the right right-count
- ④ possible ans = left-count + 1 + right-count  
→ ans = max (ans, possible\_ans)

ex:



ex: 1 1 1 1  $\Rightarrow$  4

Edge case : No zeros (all 1's)  $\Rightarrow \underline{ans = N}$

ex: 0 0 0 0

```
int solve (A) {
```

```
    int N = A.size()
```

```
    int maxOnes = INT_MIN; (min element)
```

```
    int totalOnes = 0;
```

```
    for (int i=0; i<N; i++) {
```

```
        if (A[i] == 1) {
```

```
            totalOnes++;
```

```
}
```

// there are no 0's in the array

```

if (totalOnes == N) return N;
for (int i=0; i<N; i++) {
    if (A[i] == 0) {
        j = i-1;
        l = 0; // i's on the left side
        while (j >= 0 and A[j] == 1) {
            l++;
            j--;
        }
        maxOnes = max(maxOnes, (r+l));
    }
}
return maxOnes;

```

}

ex:

①

| | | |

= 1 access for each element

②

, | O | = |

≤ 2 times access

for each element

ex:

O | | | | O | | | O | | | | O | | | | O | | | |

— — — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — — — — — — — — —

⇒ Every element is getting accessed maximum  
3 times!

∴ iteration  $\rightarrow 3N$

Directi

T.C. =  $O(3N) \Rightarrow O(N)$



what about all 0



Q. Given an array of 1's and 0's. we can swap one of 0 with a 1. Return the count of max consecutive 1's in the array. Amazon,  
Dell, MS,  
Adobe

Quiz:  $A = [1, 1, 0, 1, 1, 1]$

ex:  $\text{swap}(0, 2) \Rightarrow [0, 1, 1, 1, 1, 1] \Rightarrow 5$

$\text{swap}(1, 2) \Rightarrow [1, 0, 1, 1, 1, 1] \Rightarrow 4$

$\text{swap}(2, 3) \Rightarrow [1, 1, 1, 0, 1, 1] \Rightarrow 3$

$\{1, 1, 0, 1, 1, 1\}$

ex:

$\rightarrow [0, 1, 1, 1, 1, 1]$   
 $[1, 1, 1, 1, 1, 0]$

$A = [1 \ 1 \ \cancel{0} \ 1 \ 1 \ \cancel{0} \ 1 \ 1 \ 1]$

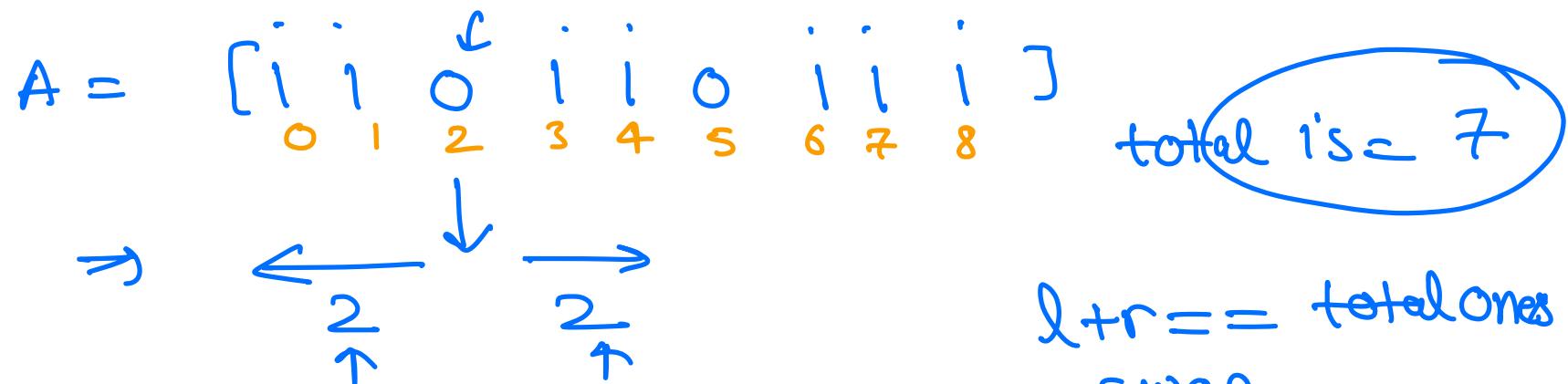
$ans = 6$

$$= [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$$

$$[1 \ 0 \ 0 \ \underline{1 \ 1 \ 1 \ 1 \ 1}] = 6$$

- idea!
- ① when we get 0 0
  - ② check 1's on the left & right  
replacing  $\rightarrow l+r+1$   
swap  $\rightarrow l+r$

ex:  $A = [1 \ 1 \ 0 \ 1 \ 1] \ ans = 4$



total count of 1's

$l+r$

$l+r+1$

$l+r == \text{total ones}$   
swap

$l+r < \text{total one}$   
 $l+r+1$

// 1. calculate total ones

// 2. for each 0's in the array

3. Count 1's on the left

4. Count 1's on the right

5. if ( $l+r == \text{total ones}$ )

$ans = \max(0, l+r)$

6.  $\text{gt}$  ( $l+r < \text{total ones}$ )

$$a_m = \max(a_m, l+r+1)$$

Q. Number of triplets:

Given an array, count the number of triplets  
 $i, j, \& k$ , such that  $i < j < k$  and  
 $A[i] < A[j] < A[k]$

$$A = [3, 9, 11, 2, 3, 12]$$

$i$	$j$	$k$	$A[i]$	$A[j]$	$A[k]$	
0	1	2	3	9	11	✓
0	1	3	3	9	2	✗

(3, 9, 11), (3, 9, 12), (3, 11, 12)

(9, 11, 12)

Count  $\Rightarrow$  4

Quiz

$$A = [2, 6, 9, 4, 10]$$

triplets = 5

i	j	k	A[i]	A[j]	A[k]	
0	1	2	2	6	9	✓
0	1	3	2	6	4	✗
0	1	4	2	< 6	10	✓
0	2	3	2	9	4	✗
0	2	4	2	9	10	✓
0	3	4	2	4	10	-
1	2	3	6	9	4	✗
.	:	:	:	:	:	.

(2, 6, 9) (2, 6, 10), (2, 4, 10), (2, 9, 10)  
(6, 9, 10)

idea: Brute force: Explore all the possibilities.

10:35

① check all possible triplets  $i, j, k$   
where  $i < j < k$

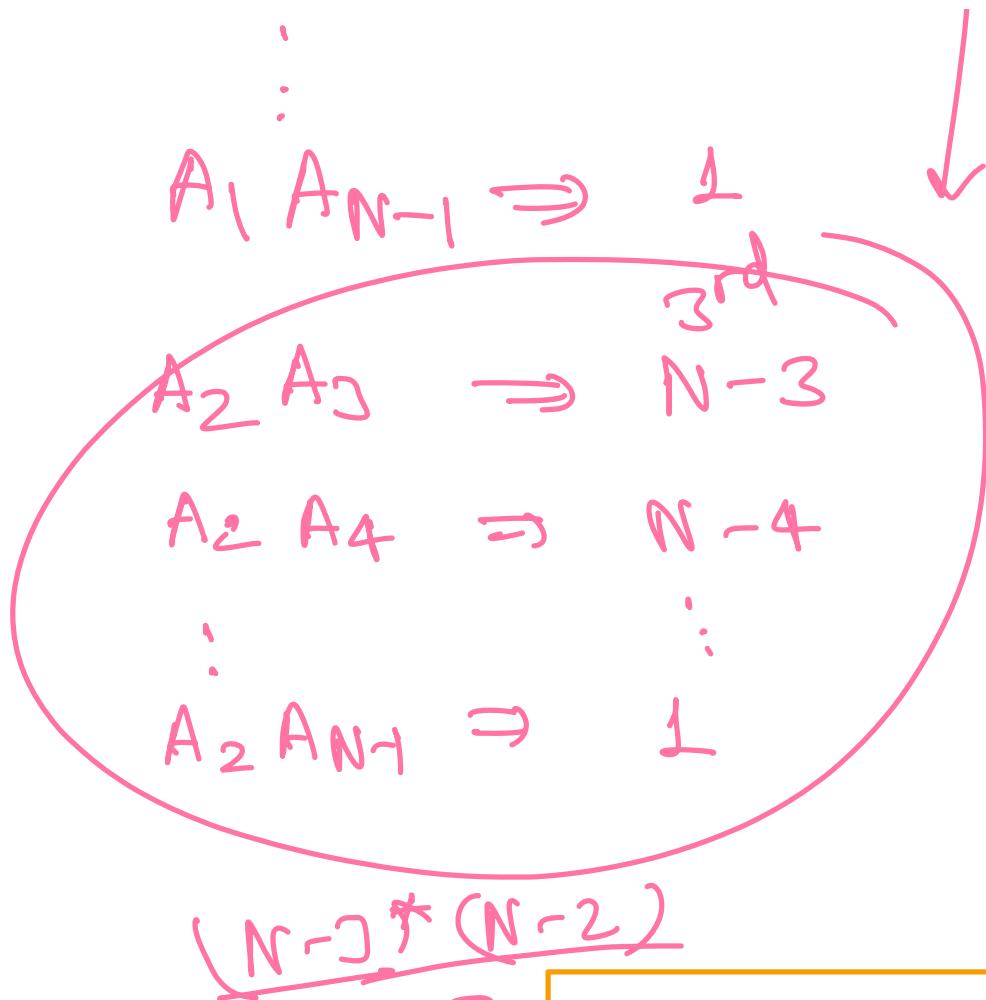
② if  $A[i] < A[j]$  and  $A[j] < A[k]$   
 $\text{count}++;$

$$\text{T.C.} = O(N^3), \text{ S.C.} = O(1)$$

Optimal sol<sup>n</sup>:

Obs<sup>n</sup>: How many triplets for <sup>Array</sup> size  $N$   
at index

$$\begin{array}{l} [A_1 \ A_2 \ A_3 \ \dots \ A_N] \\ A_1, A_2 \Rightarrow \frac{N-2}{2} \\ A_1, A_3 \Rightarrow N-3 \\ A_1, A_4 \Rightarrow \frac{N-4}{2} \end{array} \quad \left| \quad \begin{array}{l} A_1 \\ A_2 \\ \vdots \\ A_{N-1} \\ A_N \end{array} \right. \quad \Rightarrow \frac{(N-2)*(N-1)}{2}$$



$$\Rightarrow \sum_{i=1}^{N-2} \frac{(N-i) * (N-i+1)}{2}$$

$$\Rightarrow O(\underline{\underline{N^3}})$$

$\vdots$   
 $A_3 A_4 \Rightarrow \frac{N-4}{2}$   
 $A_3 A_5 \Rightarrow \frac{N-5}{2}$

$$\frac{(N-4) * (N-5)}{2}$$

```
int solve (A) {
```

```
    int N = A.size()
```

```
    int count = 0;
```

```
    for (i= 0; i < N; i++) {
```

```
        for (j= i+1; j < N; j++) {
```

```
            for (k= j+1; k < N; k++) {
```

```
                if (A[i] < A[j] < A[k]) {
```

```
                    count++
```

```
                }
```

```
    return count
```

```
}
```

$$A = \begin{bmatrix} 2, 6, 9, 10 \\ 0, 1, 2, 3 \end{bmatrix}$$

Dry Run & verify  
would this  
condition  
work or not.

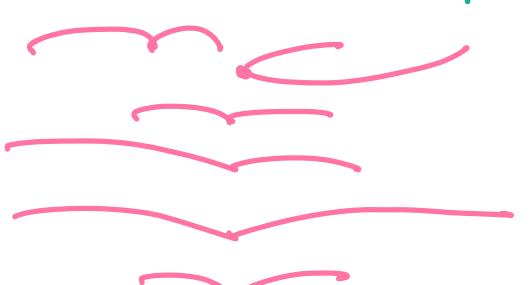
$$\text{T.C.} = O(N^3)$$
$$\text{S.C.} = O(1)$$

optimize: obs<sup>n</sup>: ① If we need to reduce the time complexity, we can not consider (check) each triplet

combination techniques:

idea: ① count how each index is contributing

$(2, 6, 9)$  -  $A = [2, 6, 9, 4, 10]$   
 $(2, 6, 10)$  -  
 $(2, 9, 10)$   
 $(2, 4, 10)$   
 $(6, 9, 10)$



{  
2 is middle = 0  
6 is middle = 2  
9 is middle = 2  
4 is middle = 1  
10 is middle = 0  
⇒ 5

idea: How many times  $A[i]$  is appearing as middle element in the triplet

Sol<sup>n</sup>: ① #times 2 is coming in the middle  
(in void triplet)

② - 6 -  
③ - 9 -  
④ - 4 -  
⑤ - 10 -

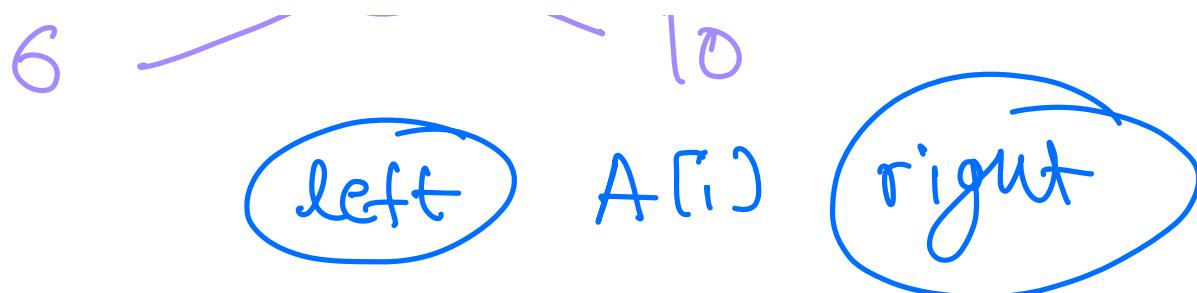
A = [2, 6, 9, 4, 10]  
      0   1   2   3   4

⇒ How many times index 2 is in middle

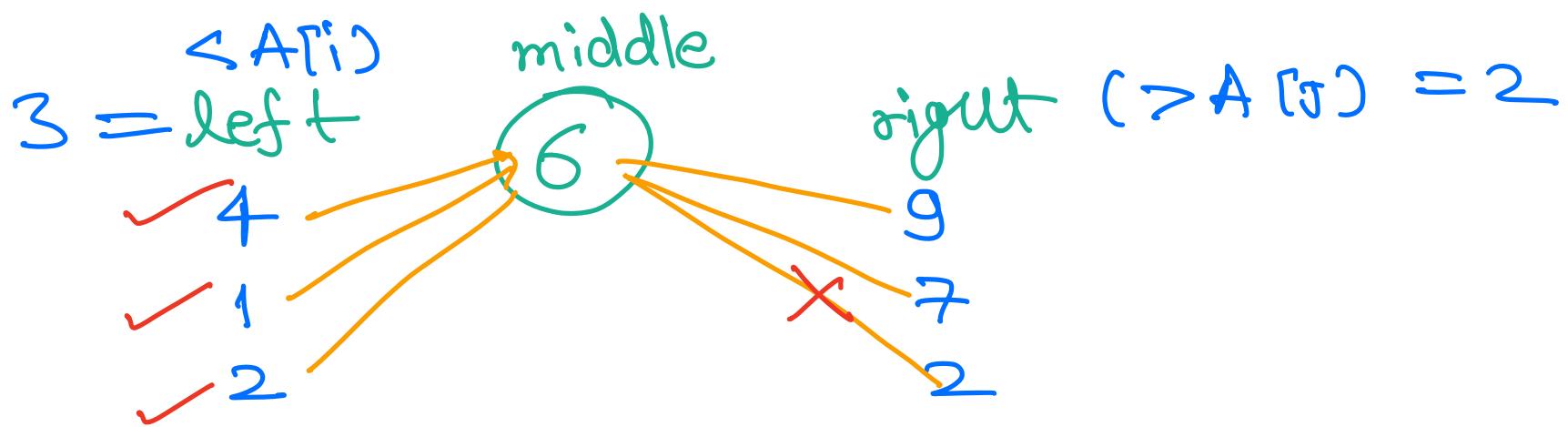
$$\frac{(0,1)}{j} \frac{2}{< j < k} \frac{(4,10)}{}$$



2, 9, 10  
6, 9, 10



$A[] = 4 \ 1 \ 2 \ 6 \ 9 \ 7 \ 2$



$< A[i]$

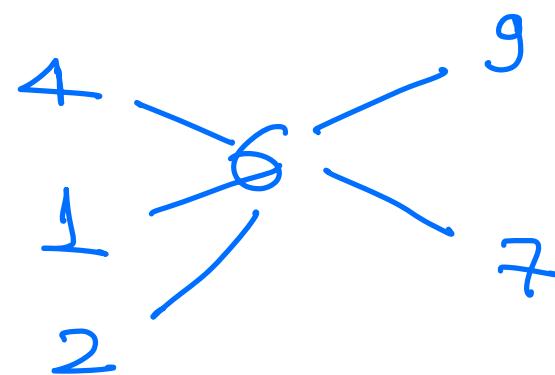
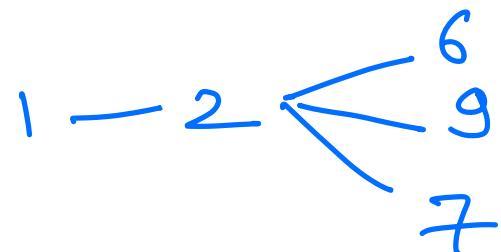
Contribution of  $A[j] =$   $(\text{left elements } < A[j]) * (\text{right elements } > A[j])$

$A[] = 4 \downarrow 1 \downarrow 2 \swarrow 6 \swarrow 9 \swarrow 7 \swarrow 2$

$l = 0 \ 0 \ 1 \ 3 \ 4 \ 4 \ 1$

$r = 3 \ 5 \ 3 \ 2 \ 0 \ 0 \ 0$

$$\text{count} = \frac{0 \ 0 \ 3 \ 6 \ 0 \ 0 \ 0}{0 \ 0 \ 3 \ 6 \ 0 \ 0 \ 0} = 9$$



int solve (A) {

    int N = A.size();

    int count = 0

```

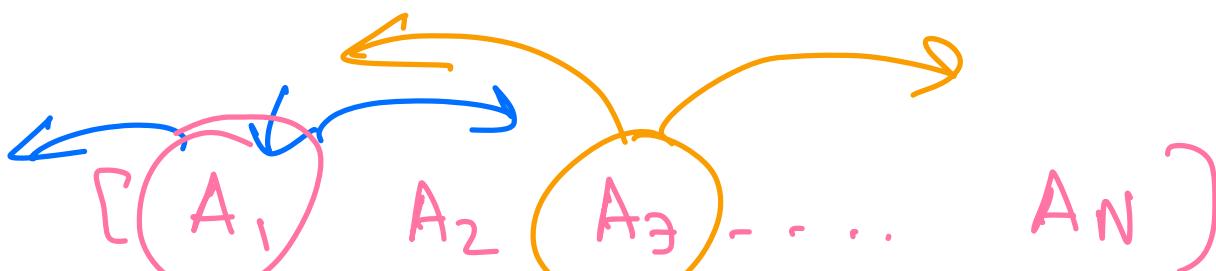
for (int j=0; j< N; j++) {
    int left=0, right=0
    int i = j-1;
    while (i ≥ 0) {
        if (A[i] < A[j]) {
            left++
            i--;
        }
    }
    int k = j+1;
    while (k < N) {
        if (A[k] > A[j] )
            right++;
        k++;
    }
    count += left*right;
}

```

$$T.C. = O(N^2)$$

b  
return count;

}



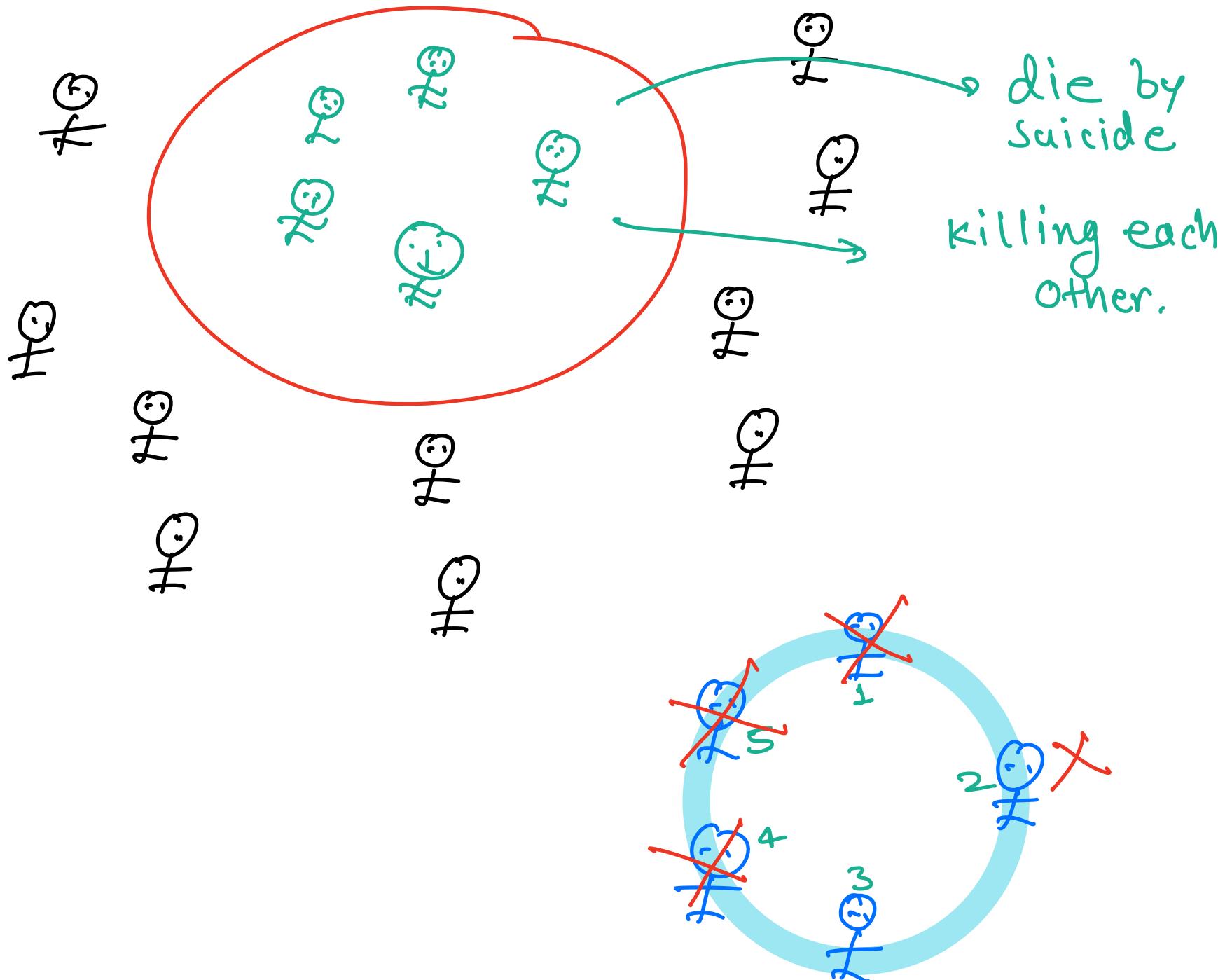
=>

visiting whole array once  $\Rightarrow O(N)$

$\Rightarrow$  visiting whole array once  $\Rightarrow O(N)$

$$T.C = \bar{N} * O(N) = \underline{\underline{O(N^2)}}$$

Q: Josephus problem:



Q.

Group of  $N$  people, standing in the circle, every next person is executed, and the sword is passed to the next person.

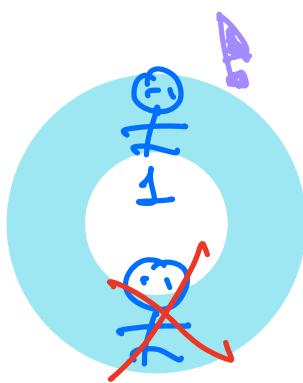


which position in the circle will be the sole survivor?

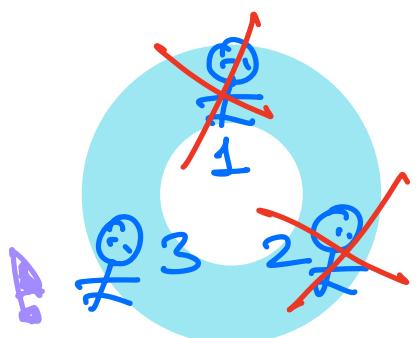
ex:



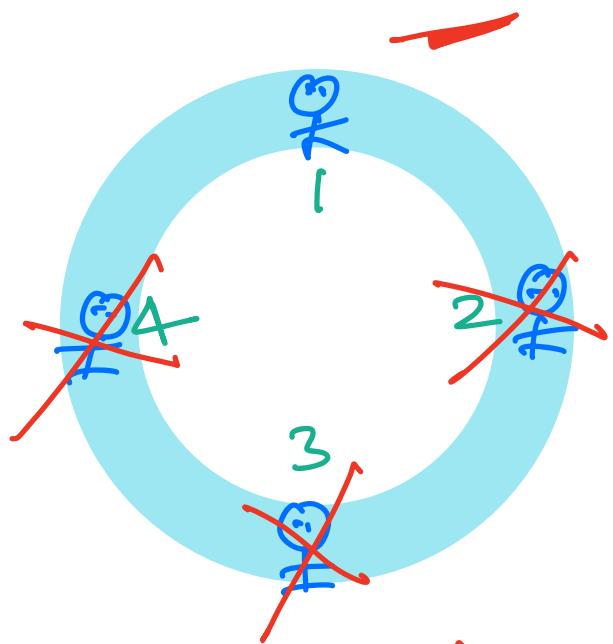
$$N=1, \text{ ans} = 1$$



$$N=2, \text{ ans} = 1$$

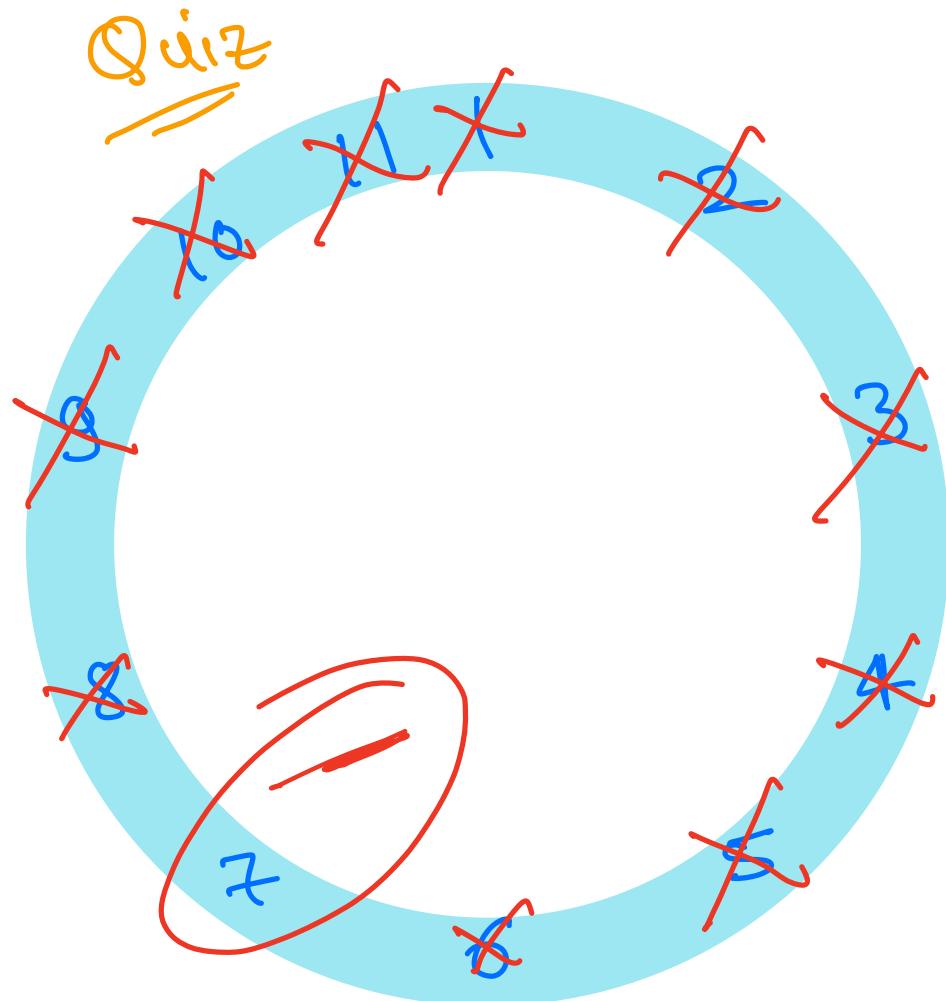
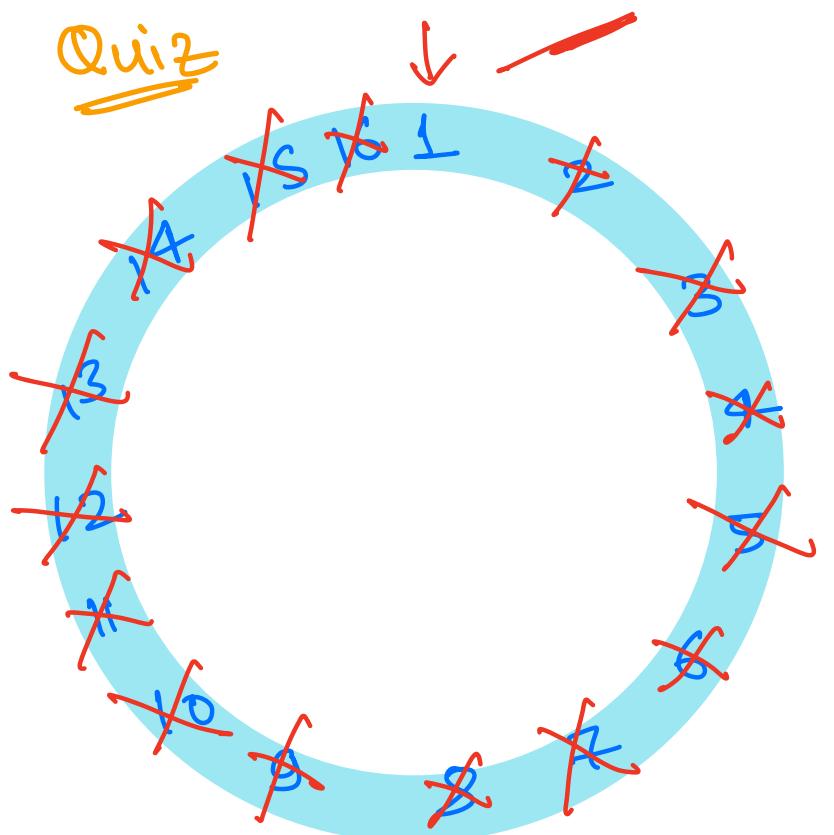


$N = 3, \text{ ans} = 3$



$N = 4, \text{ ans} = 1$

given  $N \Rightarrow$  find the survivor=?



Obs<sup>n</sup>:  
 $N = 1, 2, 4, 8, 16$

29<sup>th</sup> June  
 Extra class  
Thursday   
recorded