Lecture :- Contest Discussion 1

Agenda

- Longest strictly increasing subarray
- Negative numbers in a range
- Time complexity problem.

Qu1 **Negative numbers in a range**

Given an arr[n] and Q queries. for every query return the negative numbers in given range.

Examples: arr[] = [-2 (0) , 1 (1) , -8 (2) , 3 (3) , 9 (4) , -6 (5)]

l	r	count
0	3	2
3	5	1
1	3	1
2	4	1

Brute force:

for every query — O(Q)

Iterate the array for given range and count negative no. — O(n)

TC: O(Q * n)

Constraint:

$1 <= n <= 10^5$

$1 <= Q <= 10^5$

$Q * n = 10^{10}$ [O(Q*n) won't work]

Optimised code

	0	1	2	3	4	5
arr[] =	-2	1	-8	3	9	-6
pf[] =	1	1	2	2	2	3

pf[0] = 1 if arr[0] < 0 → 1

else → 0.

pf[1] = count of negative no [0, 1]

= pf[0] + if arr[1] < 0 → 1

else → 0

count of negative no [l, r] = pf[r] − pf[l-1]

Edge case :- l = 0

└ pf[r].

Qu Longest strictly increasing subarray.

1. arr[] = [6 9 4 8] (0 1 2 3)

arr[] = [8 4 5 | 6 11 15 19 | 1 0 5]

Brute force:

Go to all subarray — $O(n^2)$

Check if it strictly increasing or not — $O(n)$.

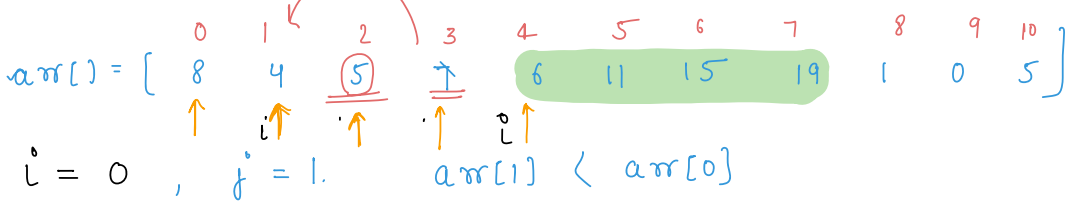
Yes — update your ans

No — Ignore.

constraint: $1 <= n <= 10^5$

$O(n^3) \simeq 10^{15}$ → not work

$O(n^2) \simeq 10^{10}$ → not work



i = 0, j = 1. arr[1] < arr[0]

i = 1

j = 2

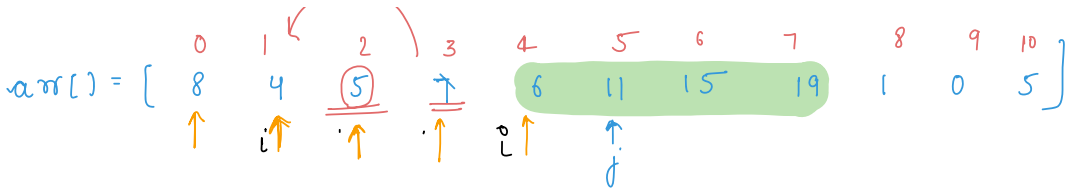
arr[j] (2) > arr[j-1] (1) → update the ans. j++

j = 3

arr[j] (3) > arr[j-1] (2) → update your ans. j++

j = 4

arr[j] > arr[j-1] → No



$i = 4$

$j = 5$

$arr[j] > arr[j-1]$

$j = 6$

$arr[6] > arr[5]$

$j = 7$

$arr[7] > arr[6]$

$j = 8$

$arr[8] < arr[7]$

$j++ \longrightarrow j - i + 1 = 2$

$j++ \longrightarrow j - i + 1 = 3$

$j++ \longrightarrow j - i + 1 = 4$

$i = 4$ and $j = 8$ [ans = $j - i$]

$i = 8$

$j = 9$

⋮

```
while( j<n){
    if( arr[j] > arr[j-1] ){
        j++;
    } else{
        break;
    }
}
```

Qu Time complexity

```
for( i=1; i<=n; i++) {
    for( j=1; j<=n; j++) {
        print (" Priyanka");
        break;
    }
    if( i* i>n) {
        break;
    }
}
```

8 * 8 = 64 > 50

n = 50

i	j [1,n]	if
1	1	1
2	1	1
3	1	1
4	1	1
!		
7	1	1
8	1	1

$n = 50$

$\sqrt{n} = 7.$

$O(\sqrt{n} + 1.) \simeq O(\sqrt{n})$

Doubts