Contest Discussion

Reload if not audible

Contest Missed Issue → will get fixed

Find Time Complexity - 12

What is the time complexity of the following code snippet

```
public static void function(int n) {
   for (int i = 1; i ⩽ n; i++) {
      for (int j = 1; j*j ⩽ n; j++) {
         print("*");
      }
      print();
   }
}
```

$\longrightarrow N$

$\rightarrow j^2 \le N$

$\downarrow$

$j \le \sqrt{N}$

$j \rightarrow \underline{[1, \sqrt{N}]}$

$\downarrow$

$\sqrt{N}$ iterations

Total = $N * \sqrt{N}$

1. $O(n^2)$
2. O(n log n)
3. O(n)
4. O(n * sqrt(n)) ← Correct

Sophie and divisibility

~~Sophie is a teacher, and she is helping her students with their math homework~~. ~~One of the problems is to~~ find the count of numbers divisible by 5 in a range. ~~To make it more interesting, she decided to ask them to~~ solve this problem for multiple ranges.

She has an array A of length N, and she needs to find the count of numbers divisible by 5 in the subarray from index l to r for every query. There are Q queries given by the array B where B[i][0] and B[i][1] gives the l and r for each query.

Problem Constraints

$1 <= N <= 10^5$

$1 <= A[i] <= 10^9$

$1 <= Q <= 10^5$

$0 <= \underbrace{B[i][0]}_{l} <= \underbrace{B[i][1]}_{r} < N$

Example

A = [5, 4, 5]

B = [[0, 1], [1, 2]]

Example

	0	1	2	3	4	5	6	7	8	9	10
A =	5	5	10	2	9	15	15	9	20	50	25
pf =	1	2	3	→ 3	3	4	5	5	6	7	8
	+1	+1			+1		→				

B

l, r		
[6 , 10]	4	$pf[10] - pf[5]$ $8 - 4 = 4$
[2 , 8]	4	$pf[8] - pf[1]$ $6 - 2 = 4$
[0 , 3]	3	$pf[3]$ 3

Important info: whether a no is divisible by 5 or not.

0-1 prefix array

arr[i] % 5 == 0 ⇒ +1

else ⇒ nothing

Prepare an array with answers of our queries.

```
                                                queries
                                              ↙
int[] sophie( int []A,  int [][]B) {
        N=  A.length
        Q  =  B.length

        // 1. Construct  prefix  array
         pf[N]
          if( A[0] % 5 ==0 )
               pf[0]  = 1
          else
               pf[0]  = 0

        for(int  i=1;  i<N;  i++) {
             if( A[i] % 5 ==0 )
                  pf[i]  = pf[i-1] +1
             else
                  pf[i] = pf[i-1]

        }
        //  Answering  the  queries  using  pf  array
         ans[Q]
        for(i=0;  i<Q ;  i++) {
             //  Answer  the  ith  query  which
             //  is  represented  as  B[i]
             //  B[i]  =  [ l, r]
              l =  B[i][0]
              r =  B[i][1]
```

```
            if  l >0:
               ans[i] =    pf[r]  -  pf[l-1]
            else
               ans[i]  =   pf[r]
      }

      return   ans;
}
```

Longest Odd Subarray

~~Samantha loves hiking and keeping track of the distances she covers.~~ She has an array A consisting of integers, and she wants to find the longest stretch where she only walked odd distances.

~~She needs your help to implement a function that can solve this problem.~~ The function should take an array of integers as input and return the length of the longest subarray where all the elements are odd.

Problem Constraints

1 <= N <= 10^5

1 <= A[i] <= 10^9

Example

A = 9 3 10 5 1 1 7 6 3

⇓

4

Example

A = 9 3 3 2 5 11 8 4 17

⇓

3

			reset					reset	
	↓	↓	↓	↓	↓	↓	↓	↓	↓
A =	9	3	10	5	1	1	7	6	3
curr_length	1	2	0	1	2	3	4	0	1
ans	1	2	2	2	2	3	4	4	4

final ans = 4

Carry Forward

ans = max (ans, curr)

```
int     longest Odd Subarray ( int [] A ) {
            N = A.length

            curr = 0
            ans = 0

            for (i=0; i<N; i++) {
                if ( A[i] % 2 == 1) {
                    curr ++
                    ans = max (ans, curr)
                }
                else {
                    curr = 0                    ← reset
                }
            }
```

```
        return ans
}
```

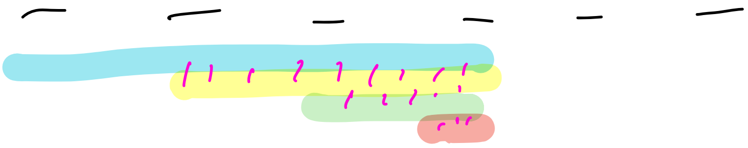
Doubts

Thank You

If you were not able to solve any question

please re-attempt the contest later on.

Q queries, [l r] range query → Prefix



Carry forward

Length of subarray

⇒ Most likely, you are not actually interested in the subarray itself.

for $\underline{i \to [0,N],}$ $\to N$

for $i \to \underline{[1,N]}$ $\to N$

for $i \to \underline{[1,N]}$ $\to N$

$3N$

$\downarrow$

$\underline{O(N)}$

Good Night

Thank You

Monday