Costly Intervals ☆

247.86 more points to get your next star!

Rank: 280176 | Points: 227.14/475

Problem Solving ★★★

Problem Submissions Leaderboard Editorial

Given an array, your goal is to find, for each element, the largest subarray containing it whose cost is at least k.

Specifically, let $A = [A_1, A_2, \ldots, A_n]$ be an array of length n, and let $A_{l\ldots r} = [A_l, \ldots, A_r]$ be the subarray from index l to index r. Also,

- Let $\mathbf{MAX}(l, r)$ be the largest number in $A_{l\ldots r}$.
- Let $\mathbf{MIN}(l, r)$ be the smallest number in $A_{l\ldots r}$.
- Let $\mathbf{OR}(l, r)$ be the bitwise OR of the elements of $A_{l\ldots r}$.
- Let $\mathbf{AND}(l, r)$ be the bitwise AND of the elements of $A_{l\ldots r}$.

The cost of $A_{l\ldots r}$, denoted $cost(l, r)$, is defined as

$$cost(l, r) = (\mathbf{OR}(l, r) - \mathbf{AND}(l, r)) - (\mathbf{MAX}(l, r) - \mathbf{MIN}(l, r)).$$

The size of $A_{l\ldots r}$ is defined as $r - l + 1$.

You are given the array A and and an integer k. For each index i from 1 to n, your goal is to find the largest size of any subarray $A_{l\ldots r}$ such that $1 \le l \le i \le r \le n$ and $cost(l, r) \ge k$.

Consider, array $A = [2, 4, 3, 1, 7]$ and $k = 6$. The possible sub-arrays and their costs would be as follows:

l,r	$A_{(l\ldots r)}$	Cost(l,r)	l,r	$A_{(l\ldots r)}$	Cost(l,r)	l,r	$A_{(l\ldots r)}$	Cost(l,r)
1,1	[2]	0	2,2	[4]	0	3,4	[3,1]	0
1,2	[2,4]	4	2,3	[4,3]	6	3,5	[3,1,7]	0
1,3	[2,4,3]	5	2,4	[4,3,1]	4	4,4	[1]	0
1,4	[2,4,3,1]	4	2,5	[4,3,1,7]	1	4,5	[1,7]	0
1,5	[2,4,3,1,7]	1	3,3	[3]	0	5,5	[7]	0

Complete the function `costlyIntervals` which takes two integers n and k as first line of input, and array $A_1, A_2, \ldots, A_n$ in the second line of input. Return an array of n integers, where the i^{th} element contains the answer for index i of the input array, $1 \le i \le n$. Every element of the output array denotes the largest size of a subarray containing i whose cost is at least k, or -1 if there is no such subarray.

Constraints

- $1 \le n \le 10^5$
- $0 \le A_i \le 10^9$

- $0 \leq k \leq 10^9$

Subtasks

- For 5% of the maximum score, $n \leq 100$.
- For 15% of the maximum score, $n \leq 5 \cdot 10^3$.

Sample Input

$n = 5, k = 6$
$A = [2, 4, 3, 1, 7]$

Sample Output

$[-1, 2, 2, -1, -1]$

Explanation

In this example, we have $k = 6$. There is only one subarray whose cost is at least 6, and that is $A_{2..3} = [4, 3]$, since $cost(2, 3) = 6$. Its size is 2. Thus, for $i = 2$ and $i = 3$, the answer is 2, and for the others, -1.

Current Buffer (saved locally, editable) Java 7

```
1  import java.io.*;
2  import java.math.*;
3  import java.security.*;
4  import java.text.*;
5  import java.util.*;
6  import java.util.concurrent.*;
7  import java.util.regex.*;
8
9  public class Solution {
10
11     // Complete the costlyIntervals function below.
12     static int[] costlyIntervals(int n, int k, int[] A) {
13         // Return a list of length n consisting of the answers
14
15     }
16
17     private static final Scanner scanner = new Scanner(System.in);
18
19     public static void main(String[] args) throws IOException {
20         BufferedWriter bufferedWriter = new BufferedWriter(new
   FileWriter(System.getenv("OUTPUT_PATH")));
21
22         String[] nk = scanner.nextLine().split(" ");
23
24         int n = Integer.parseInt(nk[0]);
25
26         int k = Integer.parseInt(nk[1]);
27
28         int[] A = new int[n];
```

```
29
30          String[] AItems = scanner.nextLine().split(" ");
31          scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
32
33          for (int i = 0; i < n; i++) {
34              int AItem = Integer.parseInt(AItems[i]);
35              A[i] = AItem;
36          }
37
38          int[] result = costlyIntervals(n, k, A);
39
40          for (int i = 0; i < result.length; i++) {
41              bufferedWriter.write(String.valueOf(result[i]));
42
43              if (i != result.length - 1) {
44                  bufferedWriter.write("\n");
45              }
46          }
47
48          bufferedWriter.newLine();
49
50          bufferedWriter.close();
51
52          scanner.close();
53      }
54  }
55
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code