

Range Minimum Query ★

Points: 1134.6100000000001 Rank: 339

Problem

Submissions

Leaderboard

RATE THIS CHALLENGE



Range Minimum Query (RMQ) is a set of problems which deals with finding a property (here minimum) of a range. Segment Tree can be very helpful when solving with such problems. A segment tree is a tree like data structure which is used to store the information about intervals. Here's the [\[wiki link\]](#) of it.

You are given a array of N integers, $arr[0], arr[1], \dots, arr[(N-1)]$. And you are given a list of ranges. For each range, (l, r) you have to find the minimum value between range $arr[l], arr[l+1], arr[l+2], \dots, arr[r]$.

Input

First line will contain two integers, N M, length of array and number of queries. Then in next line, there are N space separated integers which represent the array, $arr[0], arr[1], \dots, arr[N-1]$. Then M line follows. Each M line will contain two integers, l r, representing a range.

Output

For each range, (l, r), you have to print the minimum integer in subarray $arr[l], arr[l+1], \dots, arr[r]$ in separate line.

Constraints

 $1 \leq N, M \leq 10^5$ $-10^5 \leq arr[i] \leq 10^5$, where $0 \leq i < N$ $0 \leq l \leq r < N$

Sample Input

```
10 5
10 20 30 40 11 22 33 44 15 5
0 5
1 2
8 9
0 9
4 6
```

Sample Output

```
10
20
5
5
11
```

Explanation

- For range (0, 5), subarray will be [10, 20, 30, 40, 11, 22]. So minimum value will be 10.
- For range (1, 2), subarray will be [20, 30]. Minimum value = 20.
- For range (8, 9), subarray is [15, 5]. Minimum value = 5.
- For range (0, 9), Here we have to find the minimum (5) of the whole array.
- For range (3, 5), subarray is [40, 11, 22]. Minimum value = 11.



```
1 {-# LANGUAGE BangPatterns#-}
2
3 import Data.Array as A
4 import Control.Monad
5
6 data SegTree
7     = Empty
8     | SegNode {_value :: Int
9               , _left  :: SegTree
10              , _right :: SegTree
11              , _section :: (Int, Int)} deriving Show
12
13 buildSegTree :: A.Array Int Int -> (Int, Int) -> SegTree
14 buildSegTree arr (i,j)
15     | i == j = SegNode (arr A.! i) Empty Empty (i,i)
16     | otherwise = let l = buildSegTree arr (i, mi)
17                   r = buildSegTree arr (mi+1, j)
18                   in SegNode (min (_value l) (_value r)) l r (i,j)
19     where mi = div (i+j) 2
20
21 findMin :: SegTree -> (Int, Int) -> Int
22 findMin (SegNode v l r (i,j)) (i',j')
23     | (i,j) == (i',j') = v
24     | i' > mi = findMin r (i',j')
25     | j' <= mi = findMin l (i',j')
26     | otherwise = min (findMin l (i', mi)) (findMin r (mi+1, j'))
27     where mi = div (i+j) 2
28
```

Line: 37 Col: 1

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

