



Dashboard > Algorithms > Implementation > Service Lane

Badge Progress (Details)



Points: 1746 Rank: 15075

Service Lane

by [abhiranjan](#)

Problem

Submissions

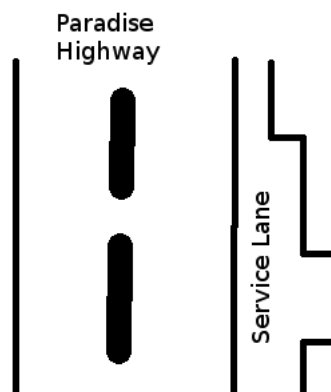
Leaderboard

Discussions

Editorial

Topics

Calvin is driving his favorite vehicle on the 101 freeway. He notices that the check engine light of his vehicle is on, and he wants to service it immediately to avoid any risks. Luckily, a service lane runs parallel to the highway. The service lane varies in width along its length.



You will be given an array of widths at points along the road (*indexes*), then a list of the indexes of entry and exit points. Considering each entry and exit point pair, calculate the maximum size vehicle that can travel that segment of the service lane safely.

Complete the code stub below to return an array of integers representing the values calculated.

Input Format

The first line of input contains two integers, n and t , where n denotes the number of width measurements you will receive and t the number of test cases. The next line has n space-separated integers which represent the array $width[w_0, w_1 \dots, w_{n-1}]$.

The next t lines contain two integers, i and j , where i is the start index and j is the end index of the segment being considered.

Constraints

- $2 \leq n \leq 100000$
- $1 \leq t \leq 1000$
- $0 \leq i < j < n$
- $2 \leq j - i + 1 \leq \min(n, 1000)$
- $1 \leq width[k] \leq 3$, where $0 \leq k < n$

Output Format

For each test case, print the number that represents the largest vehicle type that can pass through the entire segment of the service lane between indexes i and j inclusive.

Sample Input

```
8 5
2 3 1 2 3 2 3 3
0 3
4 6
6 7
```

3 5
0 7

Sample Output

1
2
3
2
1

Explanation


Below is the representation of the lane:


	HIGHWAY	Lane	->	Width
0:		--		2
1:		---		3
2:		-		1
3:		--		2
4:		---		3
5:		--		2
6:		---		3
7:		---		3


- 1. (0, 3): From index 0 through 3 we have widths 2, 3, 1 and 2. Nothing wider than 1 can pass all segments.
- 2. (4, 6): From index 4 through 6 we have widht 3, 2 and 3. Nothing wider than 2 can pass all segments.
- 3. (6, 7): 3, 3 → 3 .
- 4. (3, 5): 2, 3, 2 → 2
- 5. (0, 7): 2, 3, 1, 2, 3, 2, 3, 3 → 1.

Easy Submitted 73143 times
Max Score 20

Need Help?

 View Discussions

 View Editorial Solution

 View Top Submissions

Rate This Challenge:

☆☆☆☆☆

Resources



[Finding Max Min](#)




[Download problem statement](#)


[Download sample test cases](#)

[Suggest Edits](#)

f t in

Current Buffer (saved locally, editable)  

C++14   

1  #include <vector>

```
2 #include <algorithm>
3 #include <limits>
4 #include <iostream>
5
6 int main()
7 {
8     int n; std::cin >> n; //number of widths
9     int t; std::cin >> t; //number of test cases
10
11     std::vector<int> vec(n);
12     for (auto &it:vec) std::cin >> it;
13
14     while(t--)
15     {
16         int i, j; std::cin >> i >> j;
17         int minWidth = std::numeric_limits<int>::max();
18
19         for(int idx = i; idx <=j; ++idx)
20             minWidth = std::min(vec[idx], minWidth);
21         std::cout << minWidth << std::endl;
22     }
23 }
```

Line: 23 Col: 2

[Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

Challenge your friends: [f](#) [t](#) [in](#)

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

✓ Test Case #3

✓ Test Case #4

✓ Test Case #5

✓ Test Case #6

✓ Test Case #7

✓ Test Case #8

✓ Test Case #9

✓ Test Case #10

✓ Test Case #11

You've earned 20.00 points.

Next Challenge

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