Road Surveillance

Problem Code: SURVEIL

Design Challenge

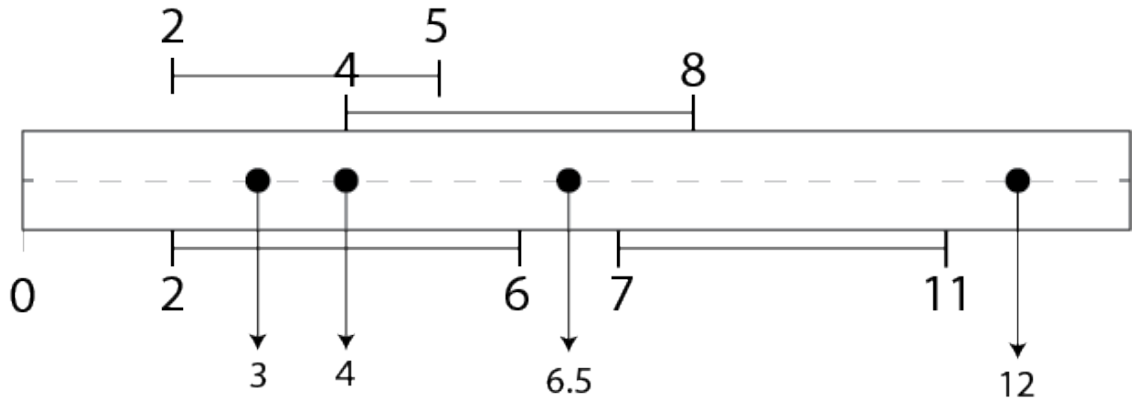
Task Description

To prevent speeding on the road the police has installed n cameras along the road. The road can be considered as the x-axis and each camera will monitor a consecutive range of the road. The monitoring range of the i-th camera is denoted by $[l_i, r_i]$ (inclusive).

Not all places along the road need a camera (there is somewhere people would never go to!). However, some places need more than one camera during peak hours. Therefore the cameras are installed in a way that ranges of the cameras may overlap. In order to check if the cameras are postioned properly, the police department needs to determine for m selected locations, how many cameras is covering that location.

The example below is a road equipped with $n = 4$ cameras. The range of them are [2, 5], [4, 8], [2, 6] and [7, 11]. $m = 4$ selected locations are identified (shown as black dots). The first location is $x_1 = 3$, 3 cameras are covering it. The 2nd, 3rd and 4th locations are $x_2 = 4, x_3 = 6.5, x_4 = 12$ and the number of cameras covering them are $3, 1, 0$ respectively.

Design an algorithm that takes the n camera ranges as input and produce the answers to m selected locations efficiently.



Constraints

$l_i < r_i$. $m = O(n)$.

Examples

Case 1:

Cameras $n = 4$, [2, 5], [4, 8], [2, 6], [7, 11]

Locations $m = 4$, [3, 4, 6.5, 12]

Answer:

As illustrated above.

Case 2:

Cameras $n = 3$, [1, 5], [2, 3], [3, 4]

Locations $m = 7$, [6, 5, 4, 3, 2, 1, 0]

Answer: [0, 1, 2, 3, 2, 1, 0]

Requirements

Time: $o(n^2)$ (little o) **Space:** $O(n)$