Problem 1893: Check if All the Integers in a Range Are Covered

Problem Information

Difficulty: Easy
Acceptance Rate: 50.71%
Paid Only: No
Tags: Array, Hash Table, Prefix Sum

Problem Description

You are given a 2D integer array `ranges` and two integers `left` and `right`. Each `ranges[i] = [starti, endi]` represents an **inclusive** interval between `starti` and `endi`.

Return `true` _if each integer in the inclusive range_ `[left, right]` _is covered by**at least one** interval in_ `ranges`. Return `false` _otherwise_.

An integer `x` is covered by an interval `ranges[i] = [starti, endi]` if `starti <= x <= endi`.

Example 1:

Input: ranges = [[1,2],[3,4],[5,6]], left = 2, right = 5 **Output:** true **Explanation:** Every integer between 2 and 5 is covered: - 2 is covered by the first range. - 3 and 4 are covered by the second range. - 5 is covered by the third range.

Example 2:

Input: ranges = [[1,10],[10,20]], left = 21, right = 21 **Output:** false **Explanation:** 21 is not covered by any range.

Constraints:

* `1 <= ranges.length <= 50` * `1 <= starti <= endi <= 50` * `1 <= left <= right <= 50`

Code Snippets

C++:

```
class Solution {
public:
bool isCovered(vector<vector<int>>& ranges, int left, int right) {

}
};
```

Java:

```
class Solution {
public boolean isCovered(int[][] ranges, int left, int right) {

}
}
```

Python3:

```
class Solution:
def isCovered(self, ranges: List[List[int]], left: int, right: int) -> bool:
```