Problem 57: Insert Interval

Problem Information

Difficulty: Medium
Acceptance Rate: 44.32%
Paid Only: No
Tags: Array

Problem Description

You are given an array of non-overlapping intervals `intervals` where `intervals[i] = [starti, endi]` represent the start and the end of the `ith` interval and `intervals` is sorted in ascending order by `starti`. You are also given an interval `newInterval = [start, end]` that represents the start and end of another interval.

Insert `newInterval` into `intervals` such that `intervals` is still sorted in ascending order by `starti` and `intervals` still does not have any overlapping intervals (merge overlapping intervals if necessary).

Return `intervals` _after the insertion_.

Note that you don't need to modify `intervals` in-place. You can make a new array and return it.

Example 1:

Input: intervals = [[1,3],[6,9]], newInterval = [2,5] **Output:** [[1,5],[6,9]]

Example 2:

Input: intervals = [[1,2],[3,5],[6,7],[8,10],[12,16]], newInterval = [4,8] **Output:** [[1,2],[3,10],[12,16]] **Explanation:** Because the new interval [4,8] overlaps with [3,5],[6,7],[8,10].

Constraints:

* `0 <= intervals.length <= 104` * `intervals[i].length == 2` * `0 <= starti <= endi <= 105` * `intervals` is sorted by `starti` in **ascending** order. * `newInterval.length == 2` * `0 <= start <= end <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> insert(vector<vector<int>>& intervals, vector<int>&
newInterval) {

}
};
```

Java:

```
class Solution {
public int[][] insert(int[][] intervals, int[] newInterval) {

}
}
```

Python3:

```
class Solution:
def insert(self, intervals: List[List[int]], newInterval: List[int]) ->
List[List[int]]:
```