

57. Insert Interval

You are given an array of non-overlapping intervals `intervals` where `intervals[i] = [starti, endi]` represent the start and the end of the th`i` 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*.

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]`.
```

Example 3:

```
Input: intervals = [], newInterval = [5,7]
Output: [[5,7]]
```

Example 4:

```
Input: intervals = [[1,5]], newInterval = [2,3]
Output: [[1,5]]
```

Example 5:

```
Input: intervals = [[1,5]], newInterval = [2,7]
Output: [[1,7]]
```

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`

```
class Solution:
    def insert(self, intervals: List[List[int]], newInterval: List[int]) ->
List[List[int]]:
        ans = []
        intervals.append(newInterval)
        intervals = sorted(intervals, key=lambda x: x[0])
        i = 0
        while i < len(intervals):
            if i == 0:
                ans.append(intervals[i])
            else:
                if ans[-1][1] >= intervals[i][0]:
                    start = ans[-1][0]
                    temp = max(ans[-1][1], intervals[i][1])
                    ans.pop()
                    ans.append([start, temp])
                else:
                    ans.append(intervals[i])
            i += 1

        return ans
```