56. Merge Intervals - Copy (1)

Given an array of `intervals` where `intervals[i] = [starti, endi]`, merge all overlapping intervals, and return *an array of the non-overlapping intervals that cover all the intervals in the input*.

Example 1:

Input: intervals = [[1,3],[2,6],[8,10],[15,18]]
Output: [[1,6],[8,10],[15,18]]
Explanation: Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].

Example 2:

Input: intervals = [[1,4],[4,5]]
Output: [[1,5]]
Explanation: Intervals [1,4] and [4,5] are considered overlapping.

Constraints:

- `1 <= intervals.length <= 104`
- `intervals[i].length == 2`
- `0 <= starti <= endi <= 104`
-

```
def merge(self, intervals: List[List[int]]) -> List[List[int]]:
        res = []
        intervals.sort(key = lambda x:x[0])
        for i in range(len(intervals)):
            if i==0:
                res.append(intervals[i])
            else:
                if intervals[i][0]<=res[-1][1]:
                    temp = res.pop()
                    temp[1] = max(temp[1],intervals[i][1])
                    res.append(temp)
                else:
                    res.append(intervals[i])
        return res
```