Problem 56: Merge Intervals

Problem Information

Difficulty: Medium
Acceptance Rate: 50.47%
Paid Only: No
Tags: Array, Sorting

Problem Description

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] overlap, 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.

Example 3:

Input: intervals = [[4,7],[1,4]] **Output:** [[1,7]] **Explanation:** Intervals [1,4] and [4,7] are considered overlapping.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> merge(vector<vector<int>>& intervals) {

}
};
```

Java:

```
class Solution {
public int[][] merge(int[][] intervals) {

}
}
```

Python3:

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