Problem 435: Non-overlapping Intervals

Problem Information

Difficulty: Medium
Acceptance Rate: 56.31%
Paid Only: No
Tags: Array, Dynamic Programming, Greedy, Sorting

Problem Description

Given an array of intervals `intervals` where `intervals[i] = [starti, endi]`, return _the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping_.

Note that intervals which only touch at a point are **non-overlapping**. For example, `[1, 2]` and `[2, 3]` are non-overlapping.

Example 1:

Input: intervals = [[1,2],[2,3],[3,4],[1,3]] **Output:** 1 **Explanation:** [1,3] can be removed and the rest of the intervals are non-overlapping.

Example 2:

Input: intervals = [[1,2],[1,2],[1,2]] **Output:** 2 **Explanation:** You need to remove two [1,2] to make the rest of the intervals non-overlapping.

Example 3:

Input: intervals = [[1,2],[2,3]] **Output:** 0 **Explanation:** You don't need to remove any of the intervals since they're already non-overlapping.

Constraints:

* `1 <= intervals.length <= 105` * `intervals[i].length == 2` * `-5 * 104 <= starti < endi <= 5 * 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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