Meeting Rooms

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Sorting
language	C++
solved on	@12/11/2024
link	https://www.geeksforgeeks.org/problems/attend-all-meetings/1
Completion	☑

Intuition

To determine if a person can attend all meetings, we need to check if there is any overlap between consecutive meetings. If any two meetings overlap, then it is impossible to attend all.

Approach

1. Sort the meetings based on their start times.
2. After sorting, check if any consecutive meetings overlap by comparing the end time of the previous meeting with the start time of the current meeting.
3. If any overlap is found, return `false`. Otherwise, return `true` after checking all consecutive pairs.

Complexity

Time Complexity:

- Sorting the meetings takes **O(n log n)**, where **n** is the number of meetings.
- Checking for overlaps takes **O(n)**.
- Overall, the time complexity is **O(n log n)**.

Space Complexity:

- The sorting operation may require additional space, depending on the sorting algorithm, but generally, it is **O(1)** for in-place sorting.

Code

```
class Solution {
  public:
    bool canAttend(vector<vector<int>> &arr) {
        sort(arr.begin(), arr.end());
        for(int i = 1; i < arr.size(); i++) {
            if(arr[i - 1][1] > arr[i][0]) return false;
        }
        return true;
```

```
    }
};
```