Problem 2054: Two Best Non-Overlapping Events

Problem Information

Difficulty: Medium
Acceptance Rate: 60.80%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Sorting, Heap (Priority Queue)

Problem Description

You are given a **0-indexed** 2D integer array of `events` where `events[i] = [startTimei, endTimei, valuei]`. The `ith` event starts at `startTimei` and ends at `endTimei`, and if you attend this event, you will receive a value of `valuei`. You can choose **at most** **two** **non-overlapping** events to attend such that the sum of their values is **maximized**.

Return _this**maximum** sum._

Note that the start time and end time is **inclusive** : that is, you cannot attend two events where one of them starts and the other ends at the same time. More specifically, if you attend an event with end time `t`, the next event must start at or after `t + 1`.

Example 1:

Input: events = [[1,3,2],[4,5,2],[2,4,3]] **Output:** 4 **Explanation:** Choose the green events, 0 and 1 for a sum of 2 + 2 = 4.

Example 2:

![Example 1 Diagram](https://assets.leetcode.com/uploads/2021/09/21/picture1.png)

Input: events = [[1,3,2],[4,5,2],[1,5,5]] **Output:** 5 **Explanation:** Choose event 2 for a sum of 5.

Example 3:

Input: events = [[1,5,3],[1,5,1],[6,6,5]] **Output:** 8 **Explanation:** Choose events 0 and 2 for a sum of 3 + 5 = 8.

Constraints:

* `2 <= events.length <= 105` * `events[i].length == 3` * `1 <= startTimei <= endTimei <= 109` * `1 <= valuei <= 106`

Code Snippets

C++:

```
class Solution {
public:
int maxTwoEvents(vector<vector<int>>& events) {

}
};
```

Java:

```
class Solution {
public int maxTwoEvents(int[][] events) {

}
}
```

Python3:

```
class Solution:
def maxTwoEvents(self, events: List[List[int]]) -> int:
```