Description | Editorial | Solutions (663) | Submissions

1751. Maximum Number of Events That Can Be Attended II

Hint

Hard 1.6K 30

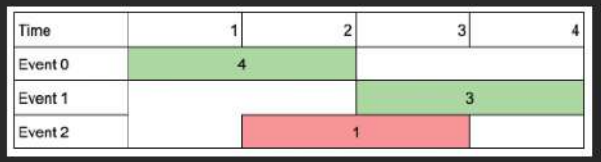
Companies

You are given an array of `events` where `events[i] = [startDay_i, endDay_i, value_i]`. The `i^th` event starts at `startDay_i` and ends at `endDay_i`, and if you attend this event, you will receive a value of `value_i`. You are also given an integer `k` which represents the maximum number of events you can attend.

You can only attend one event at a time. If you choose to attend an event, you must attend the **entire** event. Note that the end day is **inclusive**: that is, you cannot attend two events where one of them starts and the other ends on the same day.

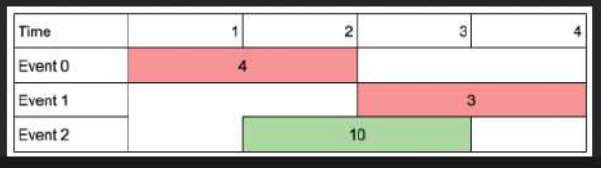
Return *the **maximum sum** of values that you can receive by attending events.*

Example 1:




```
Input: events = [[1,2,4],[3,4,3],[2,3,1]], k = 2
Output: 7
Explanation: Choose the green events, 0 and 1 (0-indexed) for a
total value of 4 + 3 = 7.
```

Example 2:



Java | Auto

```java
class Solution {
    int[][] events;
    public int maxValue(int[][] events, int k) {
        this.events = events;
        Arrays.sort(events, (a, b) -> a[0] - b[0]);
        int n = events.length;

        dp = new int[k + 1][n];
        for (int[] row : dp) {
            Arrays.fill(row, -1);
        }

        return dfs(0, k);
    }

    private int[][] dp;
    private int dfs(int index, int count) {

        if (count == 0 || index == events.length) {
```

Testcase | Result

Accepted Runtime: 1 ms

Case 1 | Case 2 | Case 3

Input

events =
`[[1,2,4],[3,4,3],[2,3,1]]`

k =
`2`

Output

`7`

Console | Run | Submit

Description | Editorial | Solutions (5.4K) | Submissions

349. Intersection of Two Arrays

Easy 4.9K 2.1K

Companies

Given two integer arrays `nums1` and `nums2`, return *an array of their intersection*. Each element in the result must be **unique** and you may return the result in **any order**.

Example 1:

```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]
```

Example 2:

```
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
Explanation: [4,9] is also accepted.
```

Constraints:

- `1 <= nums1.length, nums2.length <= 1000`
- `0 <= nums1[i], nums2[i] <= 1000`

Accepted **880.9K** | Submissions **1.2M** | Acceptance Rate **71.3%**

Seen this question in a real interview before? 1/4

Yes No

Java | Auto

```java
import java.util.Set;

class Solution {
    public int[] intersection(int[] nums1, int[] nums2) {
        Set<Integer> set1 = new HashSet<>();
        for (int i : nums1) {
            set1.add(i);
        }
        Set<Integer> set2 = new HashSet<>();
        for (int i : nums2) {
            set2.add(i);
        }
        //Removed Duplicates using Hashset
        Set<Integer> Main_Set = new HashSet<>();
        for (Integer var : set1) {
            if (set2.contains(var)) {
                Main_Set.add(var);
            }
        }
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

nums1 =
[1,2,2,1]

nums2 =
[2,2]

Output

[2]

Console | Run | Submit