Problem 1109: Corporate Flight Bookings

Problem Information

Difficulty: Medium
Acceptance Rate: 65.70%
Paid Only: No
Tags: Array, Prefix Sum

Problem Description

There are `n` flights that are labeled from `1` to `n`.

You are given an array of flight bookings `bookings`, where `bookings[i] = [firsti, lasti, seatsi]` represents a booking for flights `firsti` through `lasti` (**inclusive**) with `seatsi` seats reserved for **each flight** in the range.

Return _an array_`answer` _of length_`n` _, where_`answer[i]`_is the total number of seats reserved for flight_`i`.

Example 1:

Input: bookings = [[1,2,10],[2,3,20],[2,5,25]], n = 5 **Output:** [10,55,45,25,25] **Explanation:** Flight labels: 1 2 3 4 5 Booking 1 reserved: 10 10 Booking 2 reserved: 20 20 Booking 3 reserved: 25 25 25 25 Total seats: 10 55 45 25 25 Hence, answer = [10,55,45,25,25]

Example 2:

Input: bookings = [[1,2,10],[2,2,15]], n = 2 **Output:** [10,25] **Explanation:** Flight labels: 1 2 Booking 1 reserved: 10 10 Booking 2 reserved: 15 Total seats: 10 25 Hence, answer = [10,25]

Constraints:

* `1 <= n <= 2 * 104` * `1 <= bookings.length <= 2 * 104` * `bookings[i].length == 3` * `1 <= firsti <= lasti <= n` * `1 <= seatsi <= 104`

Code Snippets

C++:

```
class Solution {
public:
vector<int> corpFlightBookings(vector<vector<int>>& bookings, int n) {

}
};
```

Java:

```
class Solution {
public int[] corpFlightBookings(int[][] bookings, int n) {

}
}
```

Python3:

```
class Solution:
def corpFlightBookings(self, bookings: List[List[int]], n: int) -> List[int]:
```