Problem 3413: Maximum Coins From K Consecutive Bags

Problem Information

Difficulty: Medium
Acceptance Rate: 23.97%
Paid Only: No
Tags: Array, Binary Search, Greedy, Sliding Window, Sorting, Prefix Sum

Problem Description

There are an infinite amount of bags on a number line, one bag for each coordinate. Some of these bags contain coins.

You are given a 2D array `coins`, where `coins[i] = [li, ri, ci]` denotes that every bag from `li` to `ri` contains `ci` coins.

The segments that `coins` contain are non-overlapping.

You are also given an integer `k`.

Return the **maximum** amount of coins you can obtain by collecting `k` consecutive bags.

Example 1:

Input: coins = [[8,10,1],[1,3,2],[5,6,4]], k = 4

Output: 10

Explanation:

Selecting bags at positions `[3, 4, 5, 6]` gives the maximum number of coins: `2 + 0 + 4 + 4 = 10`.

Example 2:

Input: coins = [[1,10,3]], k = 2

Output: 6

Explanation:

Selecting bags at positions `[1, 2]` gives the maximum number of coins: `3 + 3 = 6`.

Constraints:

* `1 <= coins.length <= 105` * `1 <= k <= 109` * `coins[i] == [li, ri, ci]` * `1 <= li <= ri <= 109` * `1 <= ci <= 1000` * The given segments are non-overlapping.

Code Snippets

C++:

```
class Solution {
public:
long long maximumCoins(vector<vector<int>>& coins, int k) {

}
};
```

Java:

```
class Solution {
public long maximumCoins(int[][] coins, int k) {

}
}
```

Python3:

```
class Solution:
def maximumCoins(self, coins: List[List[int]], k: int) -> int:
```