Problem 2838: Maximum Coins Heroes Can Collect

Problem Information

Difficulty: Medium
Acceptance Rate: 68.66%
Paid Only: Yes
Tags: Array, Two Pointers, Binary Search, Sorting, Prefix Sum

Problem Description

There is a battle and `n` heroes are trying to defeat `m` monsters. You are given two **1-indexed** arrays of **positive** integers `heroes` and `monsters` of length `n` and `m`, respectively. `heroes[i]` is the power of `ith` hero, and `monsters[i]` is the power of `ith` monster.

The `ith` hero can defeat the `jth` monster if `monsters[j] <= heroes[i]`.

You are also given a **1-indexed** array `coins` of length `m` consisting of **positive** integers. `coins[i]` is the number of coins that each hero earns after defeating the `ith` monster.

Return _an array_`ans` _of length_`n` _where_`ans[i]`_is the**maximum** number of coins that the _`ith` _hero can collect from this battle_.

Notes

* The health of a hero doesn't get reduced after defeating a monster. * Multiple heroes can defeat a monster, but each monster can be defeated by a given hero only once.

Example 1:

Input: heroes = [1,4,2], monsters = [1,1,5,2,3], coins = [2,3,4,5,6] **Output:** [5,16,10] **Explanation:** For each hero, we list the index of all the monsters he can defeat: 1st hero: [1,2] since the power of this hero is 1 and monsters[1], monsters[2] <= 1. So this hero collects coins[1] + coins[2] = 5 coins. 2nd hero: [1,2,4,5] since the power of this hero is 4 and monsters[1], monsters[2], monsters[4], monsters[5] <= 4. So this hero collects coins[1] + coins[2] + coins[4] + coins[5] = 16 coins. 3rd hero: [1,2,4] since the power of this hero is 2 and monsters[1], monsters[2], monsters[4] <= 2. So this hero collects coins[1] + coins[2] + coins[4] = 10 coins. So the answer would be [5,16,10].

Example 2:

Input: heroes = [5], monsters = [2,3,1,2], coins = [10,6,5,2] **Output:** [23] **Explanation:** This hero can defeat all the monsters since monsters[i] <= 5. So he collects all of the coins: coins[1] + coins[2] + coins[3] + coins[4] = 23, and the answer would be [23].

Example 3:

Input: heroes = [4,4], monsters = [5,7,8], coins = [1,1,1] **Output:** [0,0] **Explanation:** In this example, no hero can defeat a monster. So the answer would be [0,0],

Constraints:

* `1 <= n == heroes.length <= 105` * `1 <= m == monsters.length <= 105` * `coins.length == m` * `1 <= heroes[i], monsters[i], coins[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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