Problem 1833: Maximum Ice Cream Bars

Problem Information

Difficulty: Medium
Acceptance Rate: 74.11%
Paid Only: No
Tags: Array, Greedy, Sorting, Counting Sort

Problem Description

It is a sweltering summer day, and a boy wants to buy some ice cream bars.

At the store, there are `n` ice cream bars. You are given an array `costs` of length `n`, where `costs[i]` is the price of the `ith` ice cream bar in coins. The boy initially has `coins` coins to spend, and he wants to buy as many ice cream bars as possible.

Note: The boy can buy the ice cream bars in any order.

Return _the**maximum** number of ice cream bars the boy can buy with _`coins` _coins._

You must solve the problem by counting sort.

Example 1:

Input: costs = [1,3,2,4,1], coins = 7 **Output:** 4 **Explanation:** The boy can buy ice cream bars at indices 0,1,2,4 for a total price of 1 + 3 + 2 + 1 = 7.

Example 2:

Input: costs = [10,6,8,7,7,8], coins = 5 **Output:** 0 **Explanation:** The boy cannot afford any of the ice cream bars.

Example 3:

Input: costs = [1,6,3,1,2,5], coins = 20 **Output:** 6 **Explanation:** The boy can buy all the ice cream bars for a total price of 1 + 6 + 3 + 1 + 2 + 5 = 18.

Constraints:

* `costs.length == n` * `1 <= n <= 105` * `1 <= costs[i] <= 105` * `1 <= coins <= 108`

Code Snippets

C++:

```
class Solution {
public:
int maxIceCream(vector<int>& costs, int coins) {

}
};
```

Java:

```
class Solution {
public int maxIceCream(int[] costs, int coins) {

}
}
```

Python3:

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