

Problem 3074: Apple Redistribution into Boxes

Problem Information

Difficulty: Easy

Acceptance Rate: 68.80%

Paid Only: No

Tags: Array, Greedy, Sorting

Problem Description

You are given an array `apple` of size `n` and an array `capacity` of size `m`.

There are `n` packs where the `i`th pack contains `apple[i]` apples. There are `m` boxes as well, and the `i`th box has a capacity of `capacity[i]` apples.

Return the **minimum** number of boxes you need to select to redistribute these `n` packs of apples into boxes.

Note that, apples from the same pack can be distributed into different boxes.

Example 1:

Input: `apple = [1,3,2]`, `capacity = [4,3,1,5,2]` **Output:** 2 **Explanation:** We will use boxes with capacities 4 and 5. It is possible to distribute the apples as the total capacity is greater than or equal to the total number of apples.

Example 2:

Input: `apple = [5,5,5]`, `capacity = [2,4,2,7]` **Output:** 4 **Explanation:** We will need to use all the boxes.

Constraints:

`1 <= n == apple.length <= 50` `1 <= m == capacity.length <= 50` `1 <= apple[i], capacity[i] <= 50` * The input is generated such that it's possible to redistribute packs of apples into boxes.

Code Snippets

C++:

```
class Solution {
public:
    int minimumBoxes(vector<int>& apple, vector<int>& capacity) {

    }
};
```

Java:

```
class Solution {
    public int minimumBoxes(int[] apple, int[] capacity) {

    }
}
```

Python3:

```
class Solution:
    def minimumBoxes(self, apple: List[int], capacity: List[int]) -> int:
```