Problem 2389: Longest Subsequence With Limited Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 73.26%
Paid Only: No
Tags: Array, Binary Search, Greedy, Sorting, Prefix Sum

Problem Description

You are given an integer array `nums` of length `n`, and an integer array `queries` of length `m`.

Return _an array_`answer` _of length_`m` _where_`answer[i]`_is the**maximum** size of a **subsequence** that you can take from _`nums` _such that the**sum** of its elements is less than or equal to _`queries[i]`.

A **subsequence** is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

Example 1:

Input: nums = [4,5,2,1], queries = [3,10,21] **Output:** [2,3,4] **Explanation:** We answer the queries as follows: - The subsequence [2,1] has a sum less than or equal to 3. It can be proven that 2 is the maximum size of such a subsequence, so answer[0] = 2. - The subsequence [4,5,1] has a sum less than or equal to 10. It can be proven that 3 is the maximum size of such a subsequence, so answer[1] = 3. - The subsequence [4,5,2,1] has a sum less than or equal to 21. It can be proven that 4 is the maximum size of such a subsequence, so answer[2] = 4.

Example 2:

Input: nums = [2,3,4,5], queries = [1] **Output:** [0] **Explanation:** The empty subsequence is the only subsequence that has a sum less than or equal to 1, so answer[0] = 0.

Constraints:

* `n == nums.length` * `m == queries.length` * `1 <= n, m <= 1000` * `1 <= nums[i], queries[i] <= 106`

Code Snippets

C++:

```
class Solution {
public:
vector<int> answerQueries(vector<int>& nums, vector<int>& queries) {

}
};
```

Java:

```
class Solution {
public int[] answerQueries(int[] nums, int[] queries) {

}
}
```

Python3:

```
class Solution:
def answerQueries(self, nums: List[int], queries: List[int]) -> List[int]:
```