

Problem 1403: Minimum Subsequence in Non-Increasing Order

Problem Information

Difficulty: Easy

Acceptance Rate: 73.35%

Paid Only: No

Tags: Array, Greedy, Sorting

Problem Description

Given the array `nums`, obtain a subsequence of the array whose sum of elements is **strictly greater** than the sum of the non included elements in such subsequence.

If there are multiple solutions, return the subsequence with **minimum size** and if there still exist multiple solutions, return the subsequence with the **maximum total sum** of all its elements. A subsequence of an array can be obtained by erasing some (possibly zero) elements from the array.

Note that the solution with the given constraints is guaranteed to be **unique**. Also return the answer sorted in **non-increasing** order.

Example 1:

Input: nums = [4,3,10,9,8] **Output:** [10,9] **Explanation:** The subsequences [10,9] and [10,8] are minimal such that the sum of their elements is strictly greater than the sum of elements not included. However, the subsequence [10,9] has the maximum total sum of its elements.

Example 2:

Input: nums = [4,4,7,6,7] **Output:** [7,7,6] **Explanation:** The subsequence [7,7] has the sum of its elements equal to 14 which is not strictly greater than the sum of elements not included ($14 = 4 + 4 + 6$). Therefore, the subsequence [7,6,7] is the minimal satisfying the conditions. Note the subsequence has to be returned in non-increasing order.

****Constraints:****

* `1 <= nums.length <= 500` * `1 <= nums[i] <= 100`

Code Snippets

C++:

```
class Solution {  
public:  
vector<int> minSubsequence(vector<int>& nums) {  
  
}  
};
```

Java:

```
class Solution {  
public List<Integer> minSubsequence(int[] nums) {  
  
}  
}
```

Python3:

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