Problem 2811: Check if it is Possible to Split Array

Problem Information

Difficulty: Medium
Acceptance Rate: 34.29%
Paid Only: No
Tags: Array, Dynamic Programming, Greedy

Problem Description

You are given an array `nums` of length `n` and an integer `m`. You need to determine if it is possible to split the array into `n` arrays of size 1 by performing a series of steps.

An array is called **good** if:

* The length of the array is **one** , or * The sum of the elements of the array is **greater than or equal** to `m`.

In each step, you can select an existing array (which may be the result of previous steps) with a length of **at least two** and split it into **two** arrays, if both resulting arrays are good.

Return true if you can split the given array into `n` arrays, otherwise return false.

Example 1:

Input: nums = [2, 2, 1], m = 4

Output: true

Explanation:

* Split `[2, 2, 1]` to `[2, 2]` and `[1]`. The array `[1]` has a length of one, and the array `[2, 2]` has the sum of its elements equal to `4 >= m`, so both are good arrays. * Split `[2, 2]` to `[2]` and `[2]`. both arrays have the length of one, so both are good arrays.

Example 2:

Input: nums = [2, 1, 3], m = 5

Output: false

Explanation:

The first move has to be either of the following:

* Split `[2, 1, 3]` to `[2, 1]` and `[3]`. The array `[2, 1]` has neither length of one nor sum of elements greater than or equal to `m`. * Split `[2, 1, 3]` to `[2]` and `[1, 3]`. The array `[1, 3]` has neither length of one nor sum of elements greater than or equal to `m`.

So as both moves are invalid (they do not divide the array into two good arrays), we are unable to split `nums` into `n` arrays of size 1.

Example 3:

Input: nums = [2, 3, 3, 2, 3], m = 6

Output: true

Explanation:

* Split `[2, 3, 3, 2, 3]` to `[2]` and `[3, 3, 2, 3]`. * Split `[3, 3, 2, 3]` to `[3, 3, 2]` and `[3]`. * Split `[3, 3, 2]` to `[3, 3]` and `[2]`. * Split `[3, 3]` to `[3]` and `[3]`.

Constraints:

* `1 <= n == nums.length <= 100` * `1 <= nums[i] <= 100` * `1 <= m <= 200`

Code Snippets

C++:

```
class Solution {
public:
```

```
bool canSplitArray(vector<int>& nums, int m) {

}
};
```

Java:

```
class Solution {
public boolean canSplitArray(List<Integer> nums, int m) {

}
}
```

Python3:

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