Problem 1655: Distribute Repeating Integers

Problem Information

Difficulty: Hard
Acceptance Rate: 40.15%
Paid Only: No
Tags: Array, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask

Problem Description

You are given an array of `n` integers, `nums`, where there are at most `50` unique values in the array. You are also given an array of `m` customer order quantities, `quantity`, where `quantity[i]` is the amount of integers the `ith` customer ordered. Determine if it is possible to distribute `nums` such that:

* The `ith` customer gets **exactly** `quantity[i]` integers, * The integers the `ith` customer gets are **all equal** , and * Every customer is satisfied.

Return `true` _if it is possible to distribute_`nums` _according to the above conditions_.

Example 1:

Input: nums = [1,2,3,4], quantity = [2] **Output:** false **Explanation:** The 0th customer cannot be given two different integers.

Example 2:

Input: nums = [1,2,3,3], quantity = [2] **Output:** true **Explanation:** The 0th customer is given [3,3]. The integers [1,2] are not used.

Example 3:

Input: nums = [1,1,2,2], quantity = [2,2] **Output:** true **Explanation:** The 0th customer is given [1,1], and the 1st customer is given [2,2].

Constraints:

* `n == nums.length` * `1 <= n <= 105` * `1 <= nums[i] <= 1000` * `m == quantity.length` * `1 <= m <= 10` * `1 <= quantity[i] <= 105` * There are at most `50` unique values in `nums`.

Code Snippets

C++:

```
class Solution {
public:
bool canDistribute(vector<int>& nums, vector<int>& quantity) {

}
};
```

Java:

```
class Solution {
public boolean canDistribute(int[] nums, int[] quantity) {

}
}
```

Python3:

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