Problem 698: Partition to K Equal Sum Subsets

Problem Information

Difficulty: Medium
Acceptance Rate: 38.29%
Paid Only: No
Tags: Array, Dynamic Programming, Backtracking, Bit Manipulation, Memoization, Bitmask

Problem Description

Given an integer array `nums` and an integer `k`, return `true` if it is possible to divide this array into `k` non-empty subsets whose sums are all equal.

Example 1:

Input: nums = [4,3,2,3,5,2,1], k = 4 **Output:** true **Explanation:** It is possible to divide it into 4 subsets (5), (1, 4), (2,3), (2,3) with equal sums.

Example 2:

Input: nums = [1,2,3,4], k = 3 **Output:** false

Constraints:

* `1 <= k <= nums.length <= 16` * `1 <= nums[i] <= 104` * The frequency of each element is in the range `[1, 4]`.

Code Snippets

C++:

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


}
};
```

Java:

```
class Solution {
public boolean canPartitionKSubsets(int[] nums, int k) {

}
}
```

Python3:

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