416. Partition Equal Subset Sum

Given a **non-empty** array `nums` containing **only positive integers**, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

Example 1:

```
Input: nums = [1,5,11,5]
Output: true
Explanation: The array can be partitioned as [1, 5, 5] and [11].
```

Example 2:

```
Input: nums = [1,2,3,5]
Output: false
Explanation: The array cannot be partitioned into equal sum subsets.
```

```
class Solution:
    def canPartition(self, nums: List[int]) -> bool:
        resSum = sum(nums)
        if resSum%2!=0:
            return False
        m = resSum//2
        n = len(nums)
        dp = [[None]*(m+1) for i in range(n+1)]
        for i in range(n+1):
            for j in range(m+1):
                if i==0 and j==0:
                    dp[i][j]=True
                elif i==0 and j!=0:
                    dp[i][j]=False
                elif j==0:
                    dp[i][j]=True
                else:
                    tar = nums[i-1]
                    if j-tar>=0:
                        dp[i][j]= dp[i-1][j] or dp[i-1][j-tar]
                    else:
                        dp[i][j]= dp[i-1][j]
        return dp[n][m]
```