

Problem Title: Equal Sum Partition (Asked by Facebook)

Problem Statement:

You are given a **multiset** (a list that can have duplicate integers). Determine whether it can be **partitioned into two subsets such that the sum of elements in both subsets is equal**.

Company:

This problem was asked by **Facebook**.

Description:

The challenge is a variation of the **Subset Sum Problem** and a classical **Dynamic Programming** problem known as **Partition Equal Subset Sum**.

You are required to determine if the given array can be split into **two subsets** A and B such that:

$$\text{sum}(A) == \text{sum}(B)$$

Examples:

Example 1:

Input: [15, 5, 20, 10, 35, 15, 10]

Output: true

Explanation:

Subset 1: [15, 5, 10, 15, 10] → Sum = 55

Subset 2: [20, 35] → Sum = 55

Example 2:

Input: [15, 5, 20, 10, 35]

Output: false

Explanation:

Total sum = 85 → Cannot be equally split into two parts.

Input Format:

- A list of integers (can include duplicates).
 - The list may contain up to 100 elements.
-

Output Format:

- Return `true` if the set can be partitioned into two subsets with equal sum, else return `false`.
-

Constraints:

- All numbers are non-negative integers.
 - At least one number exists in the input.
-

Hints:

- If total sum is odd, it can't be split evenly.
 - Use Dynamic Programming or recursion + memoization.
-

Sample Java Code:

```
public class EqualSumPartition {
    public static boolean canPartition(int[] nums) {
        int sum = 0;
        for (int num : nums) sum += num;

        if (sum % 2 != 0) return false;
        int target = sum / 2;

        boolean[] dp = new boolean[target + 1];
        dp[0] = true;

        for (int num : nums) {
            for (int j = target; j >= num; j--) {
                dp[j] = dp[j] || dp[j - num];
            }
        }

        return dp[target];
    }

    public static void main(String[] args) {
        int[] arr1 = {15, 5, 20, 10, 35, 15, 10};
        System.out.println(canPartition(arr1)); // true

        int[] arr2 = {15, 5, 20, 10, 35};
        System.out.println(canPartition(arr2)); // false
    }
}
```

References:

- Leetcode: [Partition Equal Subset Sum – Leetcode #416](#)

- GeeksForGeeks: [Partition Problem](#)
-

Video Solutions:

- [TechDose Explanation \(YouTube\)](#)
- [Dynamic Programming – Partition Equal Subset Sum](#)