

动态规划：416. 分割等和子集

```
1 class Solution {
2 public:
3     bool canPartition(vector<int>& nums) {
4         int n = nums.size();
5         int sum = 0;
6         for (auto e : nums) {
7             sum += e;
8         }
9         if (sum % 2 == 1)
10            return false;
11        int m = sum / 2;
12        vector<vector<bool>> dp(n + 1, vector<bool>(m+1, false));
13        for(int i = 0; i <= n; i++)
14        {
15            dp[i][0] = true;
16        }
17
18        for (int i = 1; i <= n; i++) {
19            for (int j = 1; j <= m; j++) {
20                dp[i][j] = dp[i - 1][j];
21                if (j >= nums[i - 1]) {
22                    dp[i][j] = (dp[i][j] || dp[i - 1][j - nums[i - 1]]);
23                }
24            }
25        }
26        return dp[n][m];
27    }
28 };
```

空间优化版本：

```
1 class Solution {
2 public:
3     bool canPartition(vector<int>& nums) {
4         int n = nums.size();
5         int sum = 0;
6         for (auto e : nums) {
7             sum += e;
8         }
9         if (sum % 2 == 1)
10            return false;
11        int m = sum / 2;
12        vector<bool> dp(m + 1, false);
13        dp[0] = true;
14        for (int i = 1; i <= n; i++) {
15            for (int j = m; j >= nums[i - 1]; j--) {
16                dp[j] = dp[j] || dp[j - nums[i - 1]];
17            }
18        }
19        return dp[m];
20    }
21 };
```