

# 416.分割等和子集

算法

动态规划

0/1背包问题

题目：Partition Equal Subset Sum

英文版链接：<https://leetcode.com/problems/partition-equal-subset-sum/description/>

中文版链接：<https://leetcode-cn.com/problems/partition-equal-subset-sum/>

## 题目分析

0/1背包问题，同样是一个非常经典的动态规划问题，也是面试常考的一类问题。

对于本题来讲，题意是非常好理解的，可以看成是一个背包大小为  $\text{sum}(\text{nums})/2$  的 0-1 背包问题。

我们来看看示例：

示例 1：

输入：[1, 5, 11, 5]

输出：true

解释：数组可以分割成 [1, 5, 5] 和 [11]。

总容量为 22，可以把它看成一个背包总容量为11的背包问题。

所以我们可以初始化一个数组w，w[i]表示能不能将背包容量填充到i，比如w[5]表示是不是能够在背包中找到元素，使得5被填满，显然是可以的。

```
w=[false,false,false,false,false,false,false,false,false,false,false]
```

我们以w[11]为例，想要将w[11]填满，有两种方式，一种是直接把11放进去，一种是通过把1 5 5 放进去。

最后只要w[11]是true，就表示这个示例输入数组能够通过某种组合，让我们的容量实现等分，也就是说能够分割等和子集。

所以我们能够得到这样一个递推式：

$$w[i] = w[i] \parallel w[i - \text{num}]$$

这样的递推式如果看不清楚，索性我们写出来代码：

```
for num in nums:
```

```
for i in range(c, num - 1, -1):
    w[i] = w[i] or w[i - num]
```

然后我们稍微按照示例推一下就很清楚了。

对于输入[1,5,11,5]来说,

当num=1时, 通过递推式只能得到w[1]=true

当num=5时, 通过递推式能够得到w[5]=true,w[6]=true, 因为可以通过1+5组合

当num=5时, 通过递推式能够得到新的w[11]=true (5+6=11)

当num=11时, 没有新改动w

所以此时可以发现w[11]=true, 所以可以等分。

## 答案

```
class Solution:
    def canPartition(self, nums) -> bool:
        # 计算总价值
        c = sum(nums)
        # 奇数直接排除
        if c % 2 != 0:
            return False
        c = c // 2
        w = [False] * (c + 1)
        # 第0个位置设置为true, 表示当元素出现的时候让w[i-num]为True, 也就是w[i]为True
        w[0] = True
        for num in nums:
            for i in range(c, num - 1, -1):
                w[i] = w[i] or w[i - num]

        return w[c]
```