

目标和

322-零钱兑换

```
class Solution:
    def coinChange(self, coins: List[int], amount: int) -> int:
        dp = [float('inf')] * (amount + 1)
        dp[0] = 0

        for coin in coins:
            # 比coin大的数可能可以有coin组成
            for x in range(coin, amount + 1):
                dp[x] = min(dp[x], dp[x - coin] + 1) # [1,2,5] 和 [1,5,2]是一样的
值为8, 过程可以无序

        return dp[amount] if dp[amount] != float('inf') else -1
```

416. 分割等和子集

```
class Solution:
    def canPartition(self, nums: List[int]) -> bool:
        sumAll = sum(nums)
        if sumAll % 2:
            return False
        target = sumAll // 2

        dp = [False] * (target + 1)
        dp[0] = True

        for i in range(len(nums)):
            for j in range(target, nums[i] - 1, -1): # 不可以使用重复数字
                dp[j] = dp[j] or dp[j - nums[i]]
        return dp[-1]
```

494--目标和

```
class Solution:
    def findTargetSumWays(self, nums: List[int], target: int) -> int:
        if target > sum(nums): return 0
        if (target+sum(nums)) & 1: return 0

        newtarget = (target+sum(nums))//2
        dp = [0] * (newtarget + 1)
        dp[0] = 1

        for i in range(len(nums)):
            for j in range(newtarget, nums[i]-1, -1): # 不可以使用重复数字
                dp[j] += dp[j - nums[i]]

        return dp[-1]
```

