

# 本日学习内容

---

1. 完成博客半篇（完全背包）
2. git三剑客一集

# 本日分享内容

---

## 题目1.

---

给你一个整数数组 `coins`，表示不同面额的硬币；以及一个整数 `amount`，表示总金额。

计算并返回可以凑成总金额所需的 **最少的硬币个数**。如果没有任何一种硬币组合能组成总金额，返回 `-1`。

你可以认为每种硬币的数量是无限的。

### 示例 1:

```
输入: coins = [1, 2, 5], amount = 11
输出: 3
解释: 11 = 5 + 5 + 1
```

### 示例 2:

```
输入: coins = [2], amount = 3
输出: -1
```

### 示例 3:

```
输入: coins = [1], amount = 0
输出: 0
```

### 提示:

- `1 <= coins.length <= 12`
- `1 <= coins[i] <= 231 - 1`
- `0 <= amount <= 104`

## 思路

`dp[i]` 表示凑出金额 `i` 所需的最少硬币数，每个硬币个数都是无限的，因此本题是一个完全背包dp。

外层遍历物品（在本题中是硬币）内层遍历背包大小，完全背包是从小到大遍历

递推公式： $dp[i] = \min(dp[i], dp[i - coin] + 1)$ ；本质上还是选不选硬币coin，在两种选择中选出较小值。思路较为简单因此很容易得出代码

## 代码

```
class Solution {
public:
    int coinChange(vector<int>& coins, int amount) {
        const int INF = amount + 1; // 初始化为一个足够大的值
        vector<int> dp(amount + 1, INF);
        dp[0] = 0; // 凑出金额 0，需要 0 个硬币

        for (int coin : coins) {
            for (int i = coin; i <= amount; ++i) {
                dp[i] = min(dp[i], dp[i - coin] + 1);
            }
        }

        return dp[amount] == INF ? -1 : dp[amount];
    }
};
```

## 题目2.

给你一个整数数组 `coins` 表示不同面额的硬币，另给一个整数 `amount` 表示总金额。

请你计算并返回可以凑成总金额的硬币组合数。如果任何硬币组合都无法凑出总金额，返回 `0`。

假设每一种面额的硬币有无限个。

题目数据保证结果符合 32 位带符号整数。

示例 1：

输入: amount = 5, coins = [1, 2, 5]

输出: 4

解释: 有四种方式可以凑成总金额:

5=5

5=2+2+1

5=2+1+1+1

5=1+1+1+1+1

### 示例 2:

输入: amount = 3, coins = [2]

输出: 0

解释: 只用面额 2 的硬币不能凑成总金额 3 。

### 示例 3:

输入: amount = 10, coins = [10]

输出: 1

## 思路

本题就是一个计数模型了, dp[i] 表示凑出金额为 i 的组合数。dp[i] 表示: 凑出金额 i 的组合数。

然后继续外层遍历物品, 内层遍历背包容器, 完全背包内层顺序便利。然后可以得出代码

## 代码

```
class Solution {
public:
    int change(int amount, vector<int>& coins) {
        int n = coins.size();
        vector<unsigned int> dp(amount + 1);
        dp[0] = 1;
        for (int coin : coins) {
            for (int j = coin; j <= amount; j++) {
                dp[j] += dp[j - coin];
            }
        }
        return dp[amount];
    }
};
```

## 本日遇到的问题

---

1. 01背包和完全背包的递推公式最初区别不清
2. 有些背包的模板题还是无法反应（砝码测重）

## 明日学习内容

---

1. 继续完全背包的博客和代码
2. git三剑客