

# 最小花费

最小花费相关的题一共两道，其中中等难度两道。

## 0322. 零钱兑换

难度：中等 **bingo!**

看了相似题目983.最低票价，很自然地也想到用动态规划解答

设dp[i]为面值为i时所需要的最少硬币数

$dp[i] = \min\{dp[i-coins[0], dp[i-coins[1]...dp[i-coins[-1]]\} + 1$

但是考虑面值为i的组合不存在，那么就设立一个最大不可达值(amount+1)

最后判断dp[-1]的结果是否可达，可达即为最少最少的硬币个数，否则返回-1

```
from typing import List

class Solution:
    def coinChange(self, coins: List[int], amount: int) -> int:
        max_size = amount + 1
        dp = [max_size] * max_size
        dp[0] = 0
        for i in range(1, max_size):
            min = dp[i]
            for j in range(len(coins)):
                if i - coins[j] >= 0:
                    if dp[i-coins[j]] + 1 < min:
                        min = dp[i-coins[j]] + 1
            dp[i] = min
        # print("max_size", max_size)
        # print(dp)
        if dp[-1] >= max_size:
            return -1
        else:
            return dp[-1]

if __name__ == '__main__':

    sol = Solution()

    coins = [1, 2, 5]
    amount = 11
    print(sol.coinChange(coins, amount)) # 3
```

```
coins = [2]
amount = 3
print(sol.coinChange(coins, amount)) # -1
```

作者: jerryqiang

链接: <https://leetcode-cn.com/problems/coin-change/solution/python3ji-bai-liao-90yi-shang-de-dai-ma-qie-tong-s/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。

## 0983. 983. 最低票价

---

难度: 中等 bingo!