322. Coin Change

Solved

Medium | Topics | Companies

You are given an integer array `coins` representing coins of different denominations and an integer `amount` representing a total amount of money.

Return *the fewest number of coins that you need to make up that amount*. If that amount of money cannot be made up by any combination of the coins, return `-1`.

You may assume that you have an infinite number of each kind of coin.

Example 1:

> **Input:** coins = [1,2,5], amount = 11
> **Output:** 3
> **Explanation:** 11 = 5 + 5 + 1

```
class Solution {
    func coinChange(_ coins: [Int], _ amount: Int) -> Int {
        if(amount == 0) {
            return 0
        }
        var dp = Array(repeating: amount+1, count:amount+1)
        dp[0] = 0

        for i in (1...amount) {
            for j in coins {
                if (i - j >= 0) {
                    dp[i] = min(dp[i],1 + dp[i-j])
                }
            }
        }

        return dp[amount] != amount+1 ? dp[amount] : -1
    }
}
```