

322 零钱兑换

Label: 动态规划

给定不同面额的硬币 `coins` 和一个总金额 `amount`。编写一个函数来计算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。

- 动态规划

```
public class Solution {
    public int coinChange(int[] coins, int amount) {
        int max = amount + 1;
        int[] dp = new int[amount + 1];
        Arrays.fill(dp, max); // 方便最后判断是否凑得齐

        dp[0] = 0;
        for (int i = 1; i < amount + 1; i++) {
            for (int j = 0; j < coins.length; j++) {
                if (coins[j] <= i) { // 这里相当于提前剪了个支，不判断这些
                    dp[i] = Math.min(dp[i], dp[i - coins[j]] + 1);
                }
            }
        }
        return dp[amount] > amount ? -1 : dp[amount];
    }
}
```