一、题目说明

题目322. Coin Change，给定一组不同面值的硬币，计算给定的总金额可以用硬币凑成的最小数量。难度是Medium！

二、我的解答

这个题目，思考了一下，和前面的`279. Perfect Squares`有点类似，属于求最优解的问题。解答方法无外乎用递归，或者dp。但是这个没做出来，由于没有找到**最优子结构**。网上找到的代码：

```
class Solution{
    public:
        //dfs + memorization
        int coinChange(vector<int>& coins,int amount){
            vector<int> memo(amount+1,-2);
            return dfs(coins,amount,memo);
        }

        int dfs(vector<int>& coins,int amount,vector<int>& memo){
            if(amount == 0) return 0;
            if(memo[amount] != -2){
                return memo[amount];
            }
            int ans = INT_MAX;
            for(int coin: coins){
                if(amount-coin<0) continue;
                int subProc = dfs(coins,amount-coin,memo);
                if(subProc == -1){
                    continue;
                }
                ans = min(ans,subProc+1);
            }
            memo[amount] = (ans == INT_MAX) ? -1 : ans;
            return memo[amount];
        }
};
```

性能：

```
Runtime: 80 ms, faster than 22.84% of C++ online submissions for Coin Change.
Memory Usage: 14.5 MB, less than 27.45% of C++ online submissions for Coin
Change.
```

三、优化措施

用dp写：

```
class Solution{
    public:
        //dp solution: dp[i] means the minimum num of coins used
        int coinChange(vector<int>& coins,int amount){
            vector<int> dp(amount+1,amount+1);
            dp[0] = 0;
```

```
            int len = coins.size();
            for(int i=1;i<=amount;i++){
                for(int j=0;j<len;j++){
                    if(coins[j]<=i){
                        dp[i] = min(dp[i],dp[i-coins[j]] + 1);
                    }
                }
            }
            if(dp[amount]==amount+1){
                return -1;
            }else return dp[amount];
        }
};
```

性能如下：

```
Runtime: 48 ms, faster than 70.77% of C++ online submissions for Coin Change.
Memory Usage: 12.6 MB, less than 86.27% of C++ online submissions for Coin
Change.
```