Problem 322: Coin Change

Problem Information

Difficulty: Medium
Acceptance Rate: 47.49%
Paid Only: No
Tags: Array, Dynamic Programming, Breadth-First Search

Problem Description

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

Example 2:

Input: coins = [2], amount = 3 **Output:** -1

Example 3:

Input: coins = [1], amount = 0 **Output:** 0

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int coinChange(vector<int>& coins, int amount) {

}
};
```

Java:

```
class Solution {
public int coinChange(int[] coins, int amount) {

}
}
```

Python3:

```
class Solution:
def coinChange(self, coins: List[int], amount: int) -> int:
```