Coin Change Problem
(Greedy Algorithm)

Coin change problem algorithm

- Sort the available coin denominations in descending order.
- Initialize a variable to keep track of the total number of coins used.
- Iterate through the coin denominations.
- While the current denomination is less than or equal to the remaining amount of change:
  - Increment the count of that denomination.
  - Subtract the value of that denomination from the remaining amount.
- Repeat until the remaining amount becomes zero.

Coin change problem example

[1,5,10,25]

- Start with the highest denomination coin, which is 25cents.
- We can use two 25 cent coins, leaving us with 63−2×25=13 cents remaining.
- Next, we move to the next highest denomination coin, which is 10 cents. We can use one 10cent coin, leaving us with 13−1×10=3 cents remaining.
- Then, we use three 1 cent coins to make up the remaining 3 cents.
- So, the greedy algorithm would use a total of 2+1+3=6 cent coins to make 63 cents in change: two 25 cent coins, one 10 cent coin, and three 1 cent coins.
- The output of the algorithm would be:
- Minimum number of coins required: 6
- Change: [25,25,10,1,1,1]