**Coin change Problem Discussion.**

Consider the following scenario:

Suppose we have a set of coins {1, 4, 6} and we want to make change for the value 8.

If we use a greedy approach, we would select the largest possible coin at each step. Starting with 6, we subtract it from 8 and get 2. We then select the largest coin less than or equal to 2, which is 1 itself. We repeat this process until we reach the value of 8.

Using this approach, we would end up with three coins: 6, 1, and 1. However, we can make change for 8 using only two coins {4, 4} which is fewer coins than the greedy approach. Hence, the greedy approach does not always give the optimal solution for the coin change problem.

If the greedy approach does not always give the optimal solution for the coin change problem, we need to use a different algorithm to find the optimal solution.

One such algorithm is the dynamic programming approach, where we break down the problem into smaller sub problems and compute the optimal solution for each sub problem. This allows us to build up the solution for larger problems.

In the coin change problem, we can use dynamic programming to find the minimum number of coins required to make change for a given value. We can do this by computing the minimum number of coins required for each value up to the target value, using the minimum number of coins required for the smaller values.

We can start with the base case of 0 coins required to make change for the value 0. We can then iterate through each coin value and compute the minimum number of coins required for each value up to the target value, using the minimum number of coins required for the smaller values. The final solution will be the minimum number of coins required for the target value.

This approach guarantees the optimal solution for the coin change problem, even in scenarios where the greedy approach does not work.