**Problem Description:**

The Coin Change problem involves finding the minimum number of coins needed to make up a given sum of money. The problem is defined by an array of coins, each with a specific denomination, and an integer representing the target sum. The goal is to minimize the number of coins used to reach the target sum.

* **Greedy Approach:**

In the greedy approach, you choose the coin with the highest denomination that is less than or equal to the remaining sum. This process is repeated until the sum becomes zero. While this approach works for some cases, it may not always produce the optimal solution. For example, if the coin denominations are not well-suited for the target sum, the greedy approach may result in using more coins than necessary.

* **Dynamic Programming:**

The dynamic programming approach involves constructing a table to store the minimum number of coins needed for each possible sum up to the target sum. The table is filled iteratively, considering all coin denominations for each sum. The final entry in the table represents the minimum number of coins needed to reach the target sum. This approach ensures that optimal subproblems are solved, leading to the overall optimal solution.

**Asymptotic Upper Bound:**

**Greedy Approach:**

- Time Complexity: O(n log n), where n is the number of coin denominations. Sorting the array of denominations dominates the time complexity.

- Space Complexity: O(1). The greedy approach does not require additional space proportional to the input size.

**Dynamic Programming:**

- Time Complexity: O(n \* targetsum), where n is the number of coin denominations and sum is the target sum. Filling the dynamic programming table requires iterating over all possible sums and coin denominations.

- Space Complexity: O(n \* targetsum). The dynamic programming table has dimensions based on the number of coin denominations and the target sum.