// e)

Given a list of coin denominations and an amount of change to be tendered, the problem is to find the minimum number of coins required to make the exact change. This is known as the coin change problem.

A greedy algorithm for the coin change problem involves selecting the largest denomination coin that can be used to make the change and subtracting it from the remaining amount until the amount becomes zero or negative. This process is repeated until the amount becomes zero.

However, the greedy algorithm may not always result in the minimum number of coins. For example, if the coin denominations are {1, 5, 10} and the amount of change to be tendered is 15, the greedy algorithm would select 10, 5 and 1, requiring a total of 3 coins. However, the optimal solution would be to use 3 coins of denomination 5, requiring a total of 3 coins.

Dynamic programming can be used to solve

the coin change problem optimally. The idea is to use a table to store the minimum number of coins required to make change for each value from 0 to the desired amount. The table is filled in a bottom-up manner, using the minimum number of coins required for smaller values to calculate the minimum number of coins required for larger values. The time complexity of the dynamic programming approach is O(amount * numCoins), where amount is the amount of change to be tendered and numCoins is the number of coin denominations.

For the given input of coin denominations {1, 7, 2, 5} and the change to be tendered of 8 and 10, the optimal solution is to use 2 and 5 coins respectively.

Bonus question:

To find the largest possible number by removing one digit from a given number N, we can iterate through each digit of N and remove it one by one to obtain a new number. The largest possible number is the maximum of all the new numbers obtained. For example, if N is 19374, the largest possible number by removing one digit is 9374.

This solution is part of a greedy algorithm because we are iteratively selecting the maximum digit to remove at each step without considering the impact of the removal on the remaining digits. It is possible that a smaller digit may need to be removed first to obtain the largest possible number. However, since the number of digits is small, the greedy approach works well in practice.