John Gill

Dr. Tala Talaei Khoei

CS 5008

5 August 2024

Assignment 3

This assignment focuses heavily on dynamic programming (DP) to solve problems. The first question had us work with the coin problems, where, given a number of coin types with different denominations, determine how many combinations of those coins can make up a total amount of money. I decided to have this solution include the ability to input the desired total so that we could run it multiple times in multiple cases. My solution was in creating a nested for loop that iterates of two variables, “i” and “j” for coin type and coin value, determining the total combinations of coins to make up the desired amount by storing each iteration of values into a table that then concatenates all the values at the end.

In question 2, I chose to solve the Fibonacci Sequence using both top-down and bottom-up approaches in dynamic programming. The top-down approach, called “memoization”, works by recursively calculating all values until we reach the base case, then concatenating them. The bottom-up approach, called “tabulation”, works by solving each case, storing each case in a temporary table, then returning the final value.

And for question 3, we had to implement a function to calculate the binomial coefficient (n choose k) using recursion and dynamic programming. My recursive solution was similar to the recursive solution implemented in the top-down approach to the last problem, where it recursively solves for n and k at decreasing values until k equals 0 or n. Then it concatenates all the values. The dynamic programming solution was also similar, in that it solves for n choose k using “i” and “j” values in a nested for loop, storing all values in a table until “j” equals 0 or 1 to trigger the base case. If the base case isn’t triggered, we use dynamic programming to solve all other values that have been stored by triggering the next descending values in the loop.