John Gill

Dr. Tala Talaei Khoei

CS 5008

5 August 2024

Assignment 3 Question 4

There are several real-world problems that can be solved using dynamic programming (DP). A first example is the knapsack problem. The knapsack problem states that, given a set of items with a specific weight and value, determine how many of each of them to include together so that the total weight is less than or equal to a given limit, then maximizes the total value, similarly to how cargo is optimized and loaded in shipping. The solution is done with dynamic programming by building a table “dp[i][w]”, with “i” being the item index, “w” being the weight limit, and the value in the table representing the maximum value possible with “i” items and “w” weight limit. Then, for each item, we determine if it can be included in the knapsack by comparing the value from including versus excluding it.

Another example that, as a Bioinformatics major, hits close to home is in string matching and sequence alignment. The problem comes in the form of comparing text strings that represent any of the sequence elements computed in Bioinformatics (DNA, RNA, or any sequence of biological elements) to find either the Longest Common Subsequence (LCS), or to measure the similarity between two sequences. The solution to the LCS is by building a table “dp[i][j]” with “i” and “j” being indices in the two sequences, and the value in the table representing the length of the LCS of their substrings. Finding the similarity between two sequences is done using Levenshtein Distance, which counts the minimum number of operations necessary for transforming one string into the other.