DSC 190 - Discussion 07

Problem 1.

Let's consider the following scenario: A thief has set their sights on robbing a house. They come prepared with a bag that has a limited capacity, denoted by W. Inside the house, there are n items, each with its own value and weight, represented in the *values* and *weights* arrays, respectively. We've previously discussed a similar problem in one of our earlier discussion sessions. However, there's a small twist to this problem compared to what we discussed previously: the thief faces a binary decision for each item. They can either choose to include an item in their bag, or they can opt to exclude it entirely. There's no middle ground where they can take a fraction of an item. The thief's ultimate goal is to maximize the total value of the stolen items. This classic problem is commonly known as the "0/1 knapsack problem".

a) Discuss why a greedy approach will not give an optimal solution for this problem. Give an example where the greedy approach would fail.

b) Write a recursive function that takes in $values, weights, index$, and W and computes the maximum value of the items that the thief can steal.

c) Optimize the above recursive solution using top-down dynamic programming.

d) Write a bottom-up dynamic programming solution for the 0/1 knapsack problem.