Since everyone will be busy with Ganesh Chaturthi (Advance Happy Ganesh Chaturthi), I have decided not to release a coding assignment this week. Instead, we will solve problems.

Longest Common Subsequence

I have worked out an example of how Dynamic Programming works for the longest common subsequence problem we saw in class.

I have worked out the details for the test case (Refer to the attached PDF):

a = “AGG”
b =”GA”

I want you to do the same for a new test case:

a = “CAT”
b = “CT”

I want you to work out the example very neatly. When you build the recursion tree, use different colors to indicate the calls and return values. I have used black and blue.

0/1 Knapsack

https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/

Refer above link for the 0/1-knapsack problem. Understand the problem, and try to come up with the solution on your own. However, it is perfectly fine to look at the solution as well.

For this problem as well, I want you to do the following:

1. Come up with a brute-force recursive solution without memory
2. Draw the recursion tree and identify the asymptotic number of nodes in the tree
3. Clearly, write where you see the redundant computations in the tree
4. Write the recursive function with memory and argue why the recursion tree corresponding to this code will have fewer nodes
5. Write the code without recursion
6. Trace the code for the for-loop variant and show the memory table

You may use the following example to solve this question:

N = 3, W = 5, profit[] = {1, 2, 3}, weight[] = {3, 2, 5}

Subset Sum

https://www.geeksforgeeks.org/subset-sum-problem-dp-25/

Subset sum is also one very famous DP question. Repeat the same for this problem as well.

You may use the test case:

set[] = {3, 6, 4, 1}, sum = 7