University of Houston

Midterm 2 Review

COSC 3320
Algorithms and Data Structures

Note

Read the Academic Honesty policy.

Exercise 1: Longest Common Subsequence (20 Points)

Let A and B be two arrays of lengths m and n, respectively. A *common subsequence* of A and B is any subsequence present in both arrays. For example, given

$$A = [3, 17, 9, 16, 9, 16, 0, 1, 6, 9]$$
$$B = [19, 10, 18, 15, 17, 7, 3, 9]$$

the sequence [17, 9] is a common subsequence of A and B. Give a dynamic programming algorithm to find the *length* of the *longest* common subsequence (LCS) of arrays A and B.

1. State the subproblems. Clearly explain your notation.
2. Give a recursive formulation to solve the subproblems.
3. Give pseudocode for your algorithm.
4. State the runtime of your algorithm. Justify your answer.
5. Explain how to modify your solution to output the actual LCS.

Exercise 2: Coin Change (20 Points)

Given coins with denominations C_1, C_2, $\ldots$, C_n and a target value t, find the minimum number of coins required to add up to t.

1. Consider the following greedy algorithm: find the coin with the greatest denomination less than or equal to t. Say that coin has denomination C. Take one such coin and repeat on $t-C$.

   Show that this algorithm does not, in general, output the optimal value.
2. Give a dynamic programming algorithm to find the minimum number of coins required to add up to t. If no combination of coins has sum t, output ∞. What is the runtime of this algorithm?

Exercise 3: Word Formation (20 Points)

Given a dictionary of words D and a string s, design a dynamic programming algorithm to determine if the string s can be broken into a sequence of words from D. What is the runtime of your algorithm? Assume that checking if a string is in the dictionary takes $\mathcal{O}(1)$ time.