CSE 312 Midterm 2

Olcay Taner YILDIZ

I. Question (Algorithm Analysis) (15 points)

Find the order of growth for solutions of the following recurrences.

a) $T(n) = 8T(n/2) + n^2$, T(1) = 1
b) $T(n) = 8T(n/2) + n^3$, T(1) = 1
c) $T(n) = 8T(n/2) + n^4$, T(1) = 1

II. Question (Divide & Conquer, Dynamic Programming) (20 points)

a) Write a recursive code for a divide-and-conquer algorithm for the exponentation problem of computing a^n where n is a positive integer.
b) Write a dynamic programming solution for the above exponentiation problem, where a^1, a^2, a^4, a^8, a^{16}, ... are stored in a dynamic programming table.

III. Question (Algorithm Analysis) (10 points)

What are the time complexities of the algorithms in Question II?

IV. Question (Divide and Conquer) (15 points)

Write a recursive pseudocode for computing the length of the longest path to a leaf node with a value of 0 in a binary tree.

V. Question (Divide and Conquer) (10 points)

Draw the binary tree with 8 nodes labeled a, b, c, d, e, f, g, h in such a way that the inorder and preorder traversals of the tree yield the following lists respectively: d, g, b, e, a, f, c (inorder); a, b, d, g, e, c, f (preorder)

VI. Question (Dynamic Programming) (10 points)

Apply the dynamic programming algorithm to the following instance of the knapsack problem:

item	weight	value
1	3	25$
2	2	20$
3	1	15$
4	4	40$
5	5	50$

VII. Question (Exhaustive Search, Dynamic Programming) (20 points)

Given an array A containing non-zero integers, determine the maximum range sum of A. In other words, find the maximum sum between two indices i and j in array A.

a) Write a brute force algorithm to solve the problem in O(n^3) time.
b) Pre-process the array A to get the dynamic programming table which contains sums from 0 to i. Using the table write the algorithm which solves the problem in O(n^2) time.