Instructor Name: Siddhartha Nath
Email: snath@scu.edu

Santa Clara University
School of Engineering
Computer Science and Engineering

Design and Analysis of Algorithms (COEN 279)
Homework 4 – 120 points

Note: For all questions, you must (i) explain your subproblems in English, (ii) provide recurrence, (iii) prove correctness, and (iv) explain and provide running time complexity to receive full credit.

Question 1 (25 points). *You are given a DAG $G = (V, E)$ with real-valued edge weights and two distinguished vertices s and t. The **weight** of a path is the sum of the weights of the edges in the path. Please describe an efficient algorithm (using dynamic programming) to find the longest weighted simple path from s to t.*

Question 2 (25 points). *A **palindrome** is a non-empty string over some alphabet that reads the same forward and backward. Examples of palindromes are all strings of length 1, civic, racecar, and aibohphobia (fear of palindromes).*
Give an efficient algorithm to find the longest palindrome that is a subsequence of a given input string of length n. For example, given the input $\{character\}$, your algorithm should return $\{carac\}$.

Question 3 (35 points). *Professor Blutarsky is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure, i.e., the supervisor relation forms a tree rooted at the president. The human resources department has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.*
Professor Blutarsky is given the tree that describes the structure of the corporation, using the left-child, right-sibling representation of a binary tree. Each node of the tree holds, in addition to the pointers, the name of an employee and that employee's conviviality ranking. Describe an efficient algorithm to make up a guest list that maximizes the sum of conviviality ratings of the guests.

Question 4 (35 points). *Suppose you are given a labelled DAG $G = (V, E, L)$, where L represents the labels on the edges. Each edge $e \in E$ has one of three possible labels– '(' (an open parenthesis), ')' (a closed parenthesis) and 0. A path in the G is said to be **balanced** if the sequence of labels of the edges along the path include the same number of open and closed parentheses. For example, paths with label sequences "(, 0 ,), (, 0,)" and "), 0, (, 0, 0" are balanced, while a path with a label sequence "(,) ,)" is not balanced.*
Given G and two specified nodes $s \in G$ and $t \in G$, design a dynamic programming algorithm to find if there exists a balanced path between s and t in G, and output the path if it exists.