Data Structures and Algorithms (CSE-102)

End-sem Exam, Date: May 3rd, 2024

Timing: 2:30 PM to 4:30 PM | Max mark: 100

Calculators are allowed to use. Use of mobile devices/ computers/ tablets is prohibited. Using books/ notes or any other reference material is not permitted. Attempt all questions; options are provided within questions, if any. Show all of your working.

1. (a) Given two sorted singly linked lists consisting of N and M nodes respectively. Write a pseudocode to merge both of the lists (in place) and return the head of the merged list. Please give a linear time algorithm. (8)

   (b) Explain Quicksort algorithm and write pseudocode. What is the best-case, and average case complexities? (8)

   (c) Write notes on Average case asymptotic time complexity of (any TWO):
   - Insert() and update() in a Max heap
   - Search in a Binary Search Tree
   - Kruskal's algorithm
   - DFS() in a graph

   (8)

2. (a) Illustrate the operation of HEAP-EXTRACT-MAX on the heap:- $A = \langle 15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1 \rangle$. (10)

   (b) Show the steps and rotations of building AVL tree from the below sequence:- 16, 15, 14, 12, 11, 8, 7, 6, 1, 99 (10)

3. (a) We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \le r(u, v) \le 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices. (36)

   OR

   (b) Show how depth-first search works on the below graph. Assume that the for loop on vertices in the algorithm considers vertices in alphabetical order, and that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex (assuming starting vertex is q). (36)

Figure 1: Graph $G = (V, E)$

4. (a) The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = (v, u) \in V \times V : (u, v) \in E$. Thus, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.

(20)

OR

(b) Given an $M \times N$ boggle board, write a pseudocode/clear steps to find a list of all possible words that can be formed by a sequence of adjacent characters on the board from a given input dictionary. We are allowed to search a word in all eight possible directions, i.e., North, West, South, East, North-East, North-West, South-East, South-West, but a word should not have multiple instances of the same cell.

Example: Consider the following the traditional 4×4 boggle board. If the input dictionary is $[START, NOTE, SAND, STONED]$, the valid words are $[NOTE, SAND, STONED]$.

M	S	E	F
R	A	T	D
L	O	N	E
K	A	F	B

Figure 2: 4×4 boggle board

(20)

Best wishes