B.E. ELECTRONICS AND TELE-COMMUNICATION ENGINEERING
SECOND YEAR SECOND SEMESTER EXAM - 2017

Data Structures And Algorithms **Time: 3 hours** **Full Marks: 100**

Answer **Q. 1** and **any five** from **the rest.**

1. Give brief answers to the following questions: **10 x 2**

(a) Define a linear linked list. How is it different from an array?
(b) Provide a pseudocode to realize the *push* operation in a stack using the linear linked list.
(c) Show that $2n^2 + 3n + 5 = O(n^2)$.
(d) Given: $T(n) = 27T\left(\frac{n}{3}\right) + n^2$. Find the asymptotic bound for $T(n)$.
(e) Define a binary tree with an example.
(f) What is a Splay tree?
(g) Define a strongly connected digraph with an example.
(h) Explain what is meant by a stable sort.
(i) Explain why the longest-common subsequence (LCS) problem cannot be solved by a simple divide-and-conquer approach.
(j) Define minimum spanning tree. Which graph search algorithm can be applied to discover a minimum spanning tree?

2. (a) Define the *insert* operation in a queue. Discuss a linked implementation of this operation.

(b) Use a stack to evaluate the postfix expression: 6, 2, 3, +, -, 3, 8, 2, /, +, *, 2, $, 3, -. Show your steps. ('$' denotes exponentiation)

(c) Use a stack to convert the infix expression: (A+B)*C to the corresponding postfix expression. Show your steps.

(2+3) + 6+ 5

3. (a) Apply the substitution method to obtain the solution of the following recurrence relation: $T(n) = T\left(\lfloor \left(\frac{n}{4}\right) \rfloor \right) + n$.

(b) Apply the recursion tree method to obtain a guess for the solution of the following recurrence relation: $T(n) = 3T\left(\frac{n}{3}\right) + n^2$.

8 + 8

4. (a) Define a binary search tree. Show the step-by-step construction of a binary search tree with the following numbers as its nodes: 65,19,74,15, 28, 69, 88, 25, 57, 80, 54.

(b) Obtain an ascending order sort of the above numbers using the binary search tree in (a).

(c) Show the effect of deleting nodes 80 and 28 from the binary search tree in (a) with proper explanations.

(2+5) + 4+ 5

[Turn over

5. (a) Argue that a complete binary tree is strictly binary but the reverse is not necessarily true.
   (b) Define balance factor and height-balanced binary search tree.
   (c) Show that the binary search tree with numbers 8, 6, 10, 4, 7, 9, 11, 3, 5, 2 as nodes is height-unbalanced. (You need not show the step-by-step construction of this tree.)
   (d) Apply AVL rotation on the tree in (c) and show that the tree becomes balanced after such operation.

3 + 4 + 4 + 5

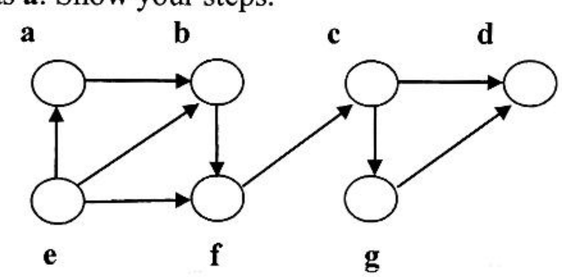
6. (a) Apply merge sort on the set of numbers: 8, 6, 10, 4, 7, 9, 11, 3, 5, 2. Show that the complexity of this process is $O(n \log n)$.
   (b) Explain the principle of binary search with an example. Show that the complexity of this process is $O(\log n)$.
   (c) Explain the problem of hash collision. Discuss one solution to tackle this problem.

6 + 5 + 5

7. (a) Mention the similarities and the differences between the dynamic programming and the greedy algorithm.
   (b) Provide a greedy algorithm based solution to fractional knapsack problem and analyze the complexity of your solution.
   (c) Obtain a dynamic programming based solution for the longest-common subsequence (LCS) problem.

4 + 6 + 6

8. (a) Find the difference in the number of edges between a complete graph of order n and a complete bipartite graph of order n.
   (b) Write a procedure for depth-first search. Apply this procedure on the following directed graph with starting node as **a**. Show your steps.



(c) Show that the parentheses structure is preserved in terms of the discovery and finishing times of visiting the nodes in (b).

4 + (3+6) + 3