ROLL NO. _______________

Code: AC64/AT64/AC115/AT115
Subject: DESIGN & ANALYSIS OF ALGORITHMS

AMIETE – CS/IT (Current & New Scheme)

Time: 3 Hours **June 2018** **Max. Marks: 100**

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE: There are 9 Questions in all.

- **Question 1 is compulsory and carries 20 marks. Answer to Q.1 must be written in the space provided for it in the answer book supplied and nowhere else.**
- **The answer sheet for the Q.1 will be collected by the invigilator after 45 minutes of the commencement of the examination.**
- **Out of the remaining EIGHT Questions answer any FIVE Questions. Each question carries 16 marks.**
- **Any required data not explicitly given, may be suitably assumed and stated.**

Q.1 Choose the correct or the best alternative in the following: (2×10)

a. To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is:
(A) Queue **(B)** Stack
(C) Heap **(D)** B-Tree

b. In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of time complexity by
(A) Dijkstra's algorithm starting from S
(B) Warshall's algorithm
(C) Performing a DFS starting from S
(D) Performing a BFS starting from S

c. An undirected graph G has n nodes. Its adjacency matrix is given by an n × n square matrix whose (i) diagonal elements are 0's and (ii) non-diagonal elements are 1's. which one of the following is TRUE?
(A) Graph G has no minimum spanning tree (MST)
(B) Graph G has a unique MST of cost n-1
(C) Graph G has multiple distinct MSTs, each of cost n-1
(D) Graph G has multiple spanning trees of different costs

d. Which of the following algorithms can be used to most efficiently determine the presence of a cycle in a given graph?
(A) Depth First Search
(B) Prim's Minimum Spanning Tree Algorithm
(C) Breadth First Search
(D) Kruskal' Minimum Spanning Tree Algorithm

e. Assuming P ≠ NP, which of the following is true?
(A) NP-complete = NP **(B)** NP-complete ∩ P = Ø
(C) NP-hard = NP **(D)** P = NP-complete

f. Suppose T(n) = 2T(n/2) + n, T(0) = T(1) = 1 Which one of the following is false?
(A) T(n) = O(n^2) **(B)** T(n) = ϴ(nLogn)
(C) T(n) = Ω(n^2) **(D)** T(n) = O(nLogn)

g. What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?
(A) Recurrence is T(n) = T(n-2) + O(n) and time complexity is O(n^2)
(B) Recurrence is T(n) = T(n-1) + O(n) and time complexity is O(n^2)
(C) Recurrence is T(n) = 2T(n/2) + O(n) and time complexity is O(nLogn)
(D) Recurrence is T(n) = T(n/10) + T(9n/10) + O(n) and time complexity is O(nLogn)

h. What is the time complexity of Build Heap operation? Build Heap is used to build a max(or min) binary heap from a given array. Build Heap is used in Heap Sort as a first step for sorting
(A) O(nLogn) **(B)** O(n^2)
(C) O(Logn) **(D)** O(n)

i. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?
i. 9679, 1989, 4199 hash to the same value
ii. 1471, 6171 hash to the same value
iii. All elements hash to the same value
iv. Each element hashes to a different value
(A) i only **(B)** ii only
(C) i and ii only **(D)** iii or iv

j. What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.
(A) 2 **(B)** 3
(C) 4 **(D)** 5

Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.

Q.2 a. Can the problem of computing the number π be solved exactly? How many instances does this problem have? **(4+4)**

b. Let A be the adjacency matrix of an undirected graph. Explain what property of the matrix indicates that (i) the graph is complete (ii) the graph has a loop, i.e., an edge connecting a vertex to itself. Also answer the above parts for the adjacency list representation. **(2+2+2+2)**

Q.3 a. For each of the following algorithms, indicate
(a) a natural size metric for its inputs;
(b) its basic operation:
i. Computing the sum of n numbers
ii. Computing n!
iii. Finding the largest element in a list of n numbers
iv. Euclid's algorithm **(2+2+2+2)**

b. The maximum values of the Java primitive types int and long are $2^{31}-1$ and $2^{63}-1$, respectively. Find the smallest n for which the n^{th} Fibonacci number is not going to fit in a memory allocated for (a) the type int (b) the type long. **(4+4)**

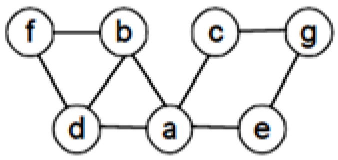
Q.4 a. Prove the equality

$$\text{lcm}(m,n) = \frac{m \cdot n}{\text{gcd}(m,n)}$$

That underlies the algorithm for computing $\text{lcm}(m,n)$. **(8)**

b. (i) Write a pseudocode for a divide-and-conquer algorithm for finding the position of the largest element in an array of n numbers.
(ii) What will be your algorithm's output for arrays with several elements of the largest value? How does this algorithm compare with the brute-force algorithm for this problem? **(4+2+2)**

Q.5 Consider the graph



a. Write down the adjacency matrix and adjacency lists specifying this graph. (Assume that the matrix rows and columns and vertices in the adjacency lists follow in the alphabetical order of the vertex labels.) **(2+2)**

b. For the above graph, starting at vertex a and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresponding depth-first search tree. Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in which the vertices became dead ends (popped off the stack). **(2+2)**

c. Apply insertion sort to sort the list E, X, A, M, P, L, E in alphabetical order. Is it possible to implement insertion sort for sorting linked lists? Will it have the same $O(n^2)$ efficiency as the array version? **(4+2+2)**

Q.6 a. Draw diagrams for the following in their general form: **(2+2)**
(i) single L-rotation and of the (ii) double RL-rotation

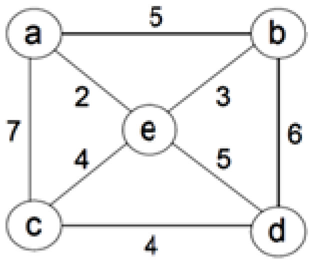
b. (i) Construct a heap for the list 1, 8, 6, 5, 3, 7, 4 by the bottom-up algorithm.
(ii) Construct a heap for the list 1, 8, 6, 5, 3, 7, 4 by successive key insertions (top-down algorithm). **(3+3)**

c. For the brute-force polynomial evaluation that stems from substituting a given value of the variable into the polynomial's formula and evaluating it from the lowest term to the highest one, determine the
(i) number of multiplications and
(ii) the number of additions made by this algorithm. **(3+3)**

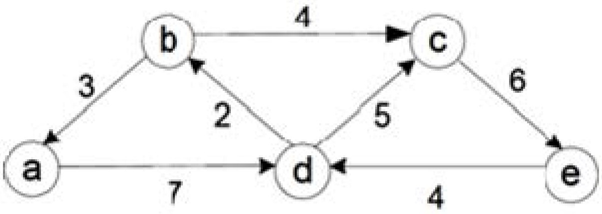
Q.7 a. What does dynamic programming have in common with divide-and conquer? What is a principal difference between the two techniques? **(2+2)**

b. Explain, why the time efficiency of Warshall's algorithm is inferior to that of the traversal-based algorithm for sparse graphs represented by their adjacency lists. **(4)**

c. Apply Prim's algorithm to the following graph. Include in the priority queue all the vertices not already in the tree. Start from vertex a. **(4)**



d. Solve the following instances of the single-source shortest-paths problem with vertex a as the source: **(4)**



Q.8 a. For the input 30, 20, 56, 75, 31, 19 and hash function h(K) = K mod 11
(i) Construct the open hash table.
(ii) Find the largest number of key comparisons in a successful search in this table.
(iii) Find the average number of key comparisons in a successful search in this table. **(3+3+2)**

b. In which context does the NP algorithms is polynomial? Give example of the running time of P and NP algorithms. **(4+4)**

Q.9 a. In context to n-queen problem, give the possible placement of queen if the queen in the table is placed at T[0,3] and draw the table. **(8)**

b. TSP: How will you show that there exists a lower bound for the travelling salesman problem? **(8)**