No of Pages : 4 **Course Code : 18XW23**

Roll No:
(To be filled in by the candidate)

PSGCOLLEGE OF TECHNOLOGY, COIMBATORE - 641 004

SEMESTER EXAMINATIONS, APRIL 2019

MSc – SOFTWARE SYSTEMS Semester : 2

18XW23 DATA STRUCTURES

Time : 3 Hours **Maximum Marks : 100**

INSTRUCTIONS:

1. Answer **ALL** questions. Each question carries 20 Marks.
2. Subdivision (a) carries 3 marks each, subdivision (b) carries 7 marks and subdivision (c) carries 10 marks each.
3. Course Outcome: Table

Qn.1	CO1	Qn.2	CO2.	Qn.3	CO3.	Qn.4	CO4	Qn.5	CO5

1. a) To sort an array, two algorithms A and B are available. Algorithm A takes (c1. n^2) and algorithm B takes (c2. n log n) running time. Algorithm A is implemented in a computer X whose execution time is 1 billion instruction per second. Algorithm B is implemented in computer Y whose execution time is 10 million instructions per second. Let a programmer code algorithm A with $2n^2$ instructions and another programmer code algorithm B in 50 n log n instructions. Find the best algorithm.

   b) (i) A lower triangular matrix is one in which all elements above the main diagonal of a square matrix are zero. Assume that we have a lower triangular matrix L with n rows as shown in Fig 1.

$$\mathbf{L} = \begin{bmatrix} l_{1,1} & & & & 0 \\ l_{2,1} & l_{2,2} & & & \\ l_{3,1} & l_{3,2} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n,1} & l_{n,2} & \dots & l_{n,n-1} & l_{n,n} \end{bmatrix}$$

Fig 1. Lower Triangular Matrix

What is the total number of nonzero terms of L? If each element can be stored in a 32- bit signed integer, how much memory would be needed to store all the nonzero elements of matrix L. [3]

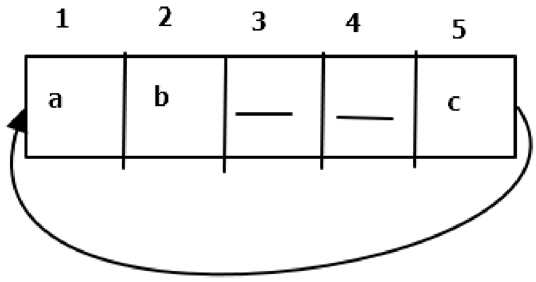
(ii) Consider the following matrix.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Why is the above matrix called sparse? Represent the above matrix using triplet notation. Assuming each integer occupies 2 bytes, find the percentage of reduction in the triplet notation. [4]

c) An array data structure has to hold the elements from 0 to n-1. Write algorithms for the operations insert, delete and search, such that the time complexity for all operations is O(1). No order on the elements (ascending or descending) is required.

2. a) The initial configuration of a circular queue is as follows.



What is the status of states of queue contents after the following sequence of steps?
enqueue x
dequeue
enqueue y
dequeue
dequeue

b) (i) Consider the method for conversion of an infix expression to its postfix form using stack. What will be the second last operator (last but one) that gets popped off the stack by the end of conversion of the following expression using this method?

(a+b)*((c+d)/(e-f)) [3]

(ii) Write an algorithm that will take two sorted stacks A and B (min on top) and create one stack that is sorted (min on top). You are allowed to use only the stack operations such as pop, push and empty. No other data structure such as arrays are allowed. [4]

c) Show how to implement a queue data structure Q using two stacks S1 and S2. Note that when you use the stack for implementing a queue, you are only allowed to use the stack commands push(x,S),pop(S) and empty(S). Also analyze the time complexity of the operations enqueue(x,Q) and deque(Q).

3. a) What does the following function do for a given linked list 1->2->3->4->5 with first node as head?

```
void fun1(struct node* head)
{
   if(head == NULL)
     return;
   fun1(head->next);
  printf("%d  ", head->data);
}
```

b) (i) Consider a sorted circular doubly-linked list where the head element points to the smallest element in the list. What is the asymptotic complexity of determining
- the smallest element in the list?
- the largest element in the list?
- whether a given element e appears in the list? [3]

(ii) Write an algorithm to insert a node into a sorted doubly linked list. [4]

c) Write algorithms to implement push, pop operations of stack and enqueue, dequeue operations of queue using singly linked list.

4. a) Draw a min heap tree that results from adding the following values, in the order given, to an initially empty heap:

7,6,5,4,3,2,1

Show the result after the deletion of the root of this heap.

b) i) A binary tree T has 9 nodes. The inorder and preorder traversals of T yield the following

Inorder traversal	:	E	A	C	K	F	H	D	B	G
Preorder traversal	:	F	A	E	K	C	D	H	G	B

Draw the binary tree. [3]

ii) Suppose that we first insert an element x into a binary search tree that does not already contain x. Suppose that we then immediately delete x from the tree. Will the new tree be identical to the original one? Defend your answer. [4]

c) Insert into an initially empty binary search tree items with the following keys (in this order): 30, 40, 23, 58, 48, 26, 11, 13. Draw the tree after each insertion. Now from the above resultant tree delete the following keys in sequence: 40, 58, 30, 23. Show your tree after each deletion. In case of deleting a key having both child follow the rule of swapping the key with left side largest.

5. a) Suppose that in a group of 5 people: A, B, C, D, and E, the following pairs of people are acquainted with each other.

A and C,

A and D,

B and C,

C and D,

C and E.

Draw a graph G to represent the above situation. Also draw an adjacency matrix for G.

b) i) Which sorting algorithm is best if the list is already in sorted order and why? [3]

ii) Sort the following sequence of numbers using radix sort.

10,21,17,34,44,11,654,123 [4]

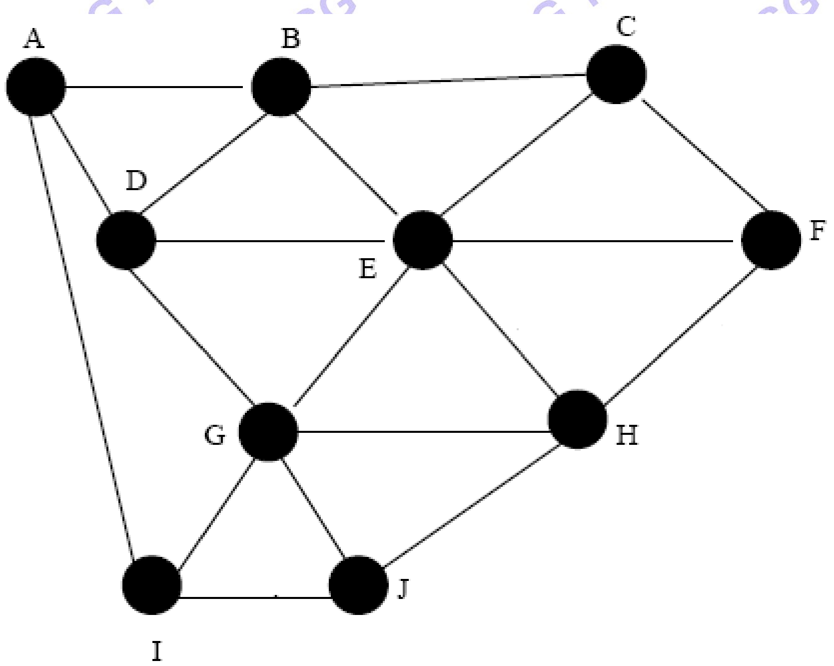
c) (i) Define Hash table. Explain different methods of forming hash functions. Insert the following data into a hash table using hash function h(X) = X mod 9. { 17, 9, 34, 56, 11, 71, 86, 55, 22, 10, 4, 39, 49, 52, 82, 13, 40, 31. 35, 28, 44} for each of the following scenarios.

- Collisions are handled by linear probing (slots = 3)
- Collisions are handled by separate chaining

(OR)

ii) For the following graph G

- Report the order of the vertices encountered on a breadth-first search starting from vertex A. Break all ties by picking the vertices in alphabetical order.
- Report the order of the vertices encountered on a depth-first search starting from vertex A. Break all ties by picking the vertices in alphabetical order.



/END/

FD/RL