DATA STRUCTURES AND ALGORITHMS
(CSE2101)

Time Allotted : 2½ hrs **Full Marks : 60**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 4 (four) from Group B to E, taking one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A

1. Answer any twelve: **12 × 1 = 12**

Choose the correct alternative for the following

(i) Let A be a two-dimensional array declared as follows:
A: array [1…10] [1…15] of character;
Assuming that each character takes one memory locations the array is stored in row-major order and the first element of the array is stored in location 100, what is the address of the element A[i][j] ?
(a) 15i + j + 84 (b) 15j + i + 84
(c) 10i + j + 89 (d) 10j + i + 89

(ii) The time complexity of inserting an element at the beginning of a singly-circular linked list having only one external pointer pointing to the head is:
(a) O (1) (b) O ($\log_2 n$) (c) O (n) (d) O (n^2)

(iii) What is the output of following function for start pointing to first node of the following linked list:
1->2->3->4->5->6?

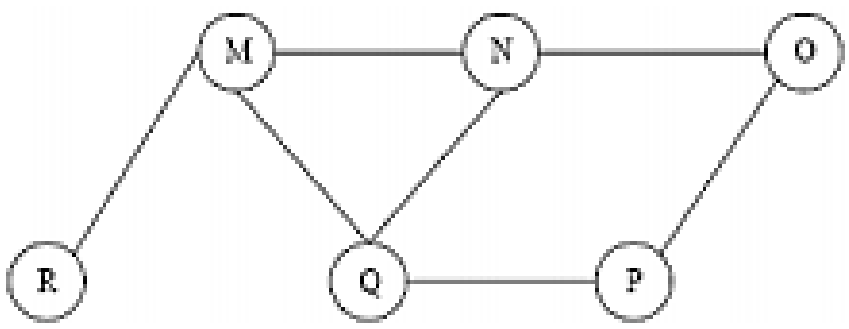
```
void fun(struct node* start)
{
    if(start == NULL)
        return;
    printf("%d ", start->data);
    if(start->next != NULL )
        fun(start->next->next);
    printf("%d ", start->data);
}
```

(a) 1 4 6 6 4 1 (b) 1 3 5 1 3 5
(c) 1 2 3 5 (d) 1 3 5 5 3 1

(iv) The equivalent Postfix notation for ((A + B) * C – (D – E) ^ (F + G)) is:
(a) AB + C * DE - - FG + ^ (b) ^ - * +ABC - DE + FG
(c) (AB)+CDE* -- FG+^ (d) AB*+DE—FG+^

(v) A stack S has the entries a,b,c with ‘a’ on the top. Another stack T is empty. An entry popped out of the Stack S can be printed immediately or pushed in to the stack T. Finally, when stack S is empty, the elements of stack T (if exists) is popped and printed. Then which sequence can never be printed?
(a) a b c (b) b a c (c) b c a (d) c a b.

(vi) The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



(a) MNOPQR (b) NQMPOR (c) QMNPRO (d) QMNPOR

(vii) The number of edges in a complete undirected graph with n vertices is
(a) n(n-l) (b) n(n-l)/2 (c) n^2 (d) 2n-1.

(viii) Which of the following is true regarding a Max Heap?
(a) Maximum element occurs at the root (b) Minimum element occurs as the last child element
(c) BuildMaxHeap has time complexity O(n log n) (d) HeapSort can have worst case time complexity as O(n^2).

(ix) Which of the following statement(/s) is (/are) correct?
Statement 1: Each B-Tree of order 2 will always be a Binary Search Tree
Statement 2: Every graph will always be a tree.
Statement 3: Every tree will be a graph, but every graph will not always be a tree
Statement 4: Every graph will be a tree, but every tree will not always be a graph.
(a) Both Statement 2 and Statement 4 (b) Only Statement 1
(c) Both Statement 1 and Statement 3 (d) Only Statement 3.

(x) If an array has n keys which are uniformly distributed, the number of comparisons required to locate a key x that is already present in the array is:
(a) 1 (b) $\sqrt{n}$ (c) n/2 (d) $\log_n x$

Fill in the blanks with the correct word

(xi) Linked lists are not suitable to implement _____ search.

(xii) A linked list having n nodes (n >= 1) where no node stores a NULL pointer is a _____ linked list.

(xiii) If the in-order traversal of a tree is E A C K F H D B G, its pre-order traversal is _____.

(xiv) The collision resolution technique where the colliding element is placed in the next available empty slot in the hash table is called _____.

(xv) The time complexity of building a max-heap is _____.

Group - B

2. (a) Propose an algorithm to remove duplicates from an ordered array, without using any second array. For example, if the original content of the array is: 1, 1, 3, 4, 8, 8, 10, 10, 25, 28, 30, 30, 30; then the final content of the array should be 1,3,4,8,10,25,28,30
What is the time complexity of the proposed algorithm? *[(CO1,CO4)(Analyze/IOCQ)]*
(b) If the number of nonzero elements in the triple representation of a sparse matrix of m × n dimension is K, then what is the time complexity to read an element from the index [i][j] of the sparse matrix. Assume that the sparse matrix is stored in triple format. *[(CO4)(Analyze/LOCQ)]*
(c) Write an algorithm/C-like pseudo code to delete the last element from a circular singly linked list. What is the time complexity of this operation? *[(CO1,CO2)(Understand/LOCQ)]*
(5 + 1) + 2 + (3 + 1) = 12

3. (a) How do you merge two sorted linked lists into a single sorted linked list? Provide an algorithm and analyze its time complexity. *[(CO3,CO4)(Analyse/HOCQ)]*
(b) State the advantages and disadvantages of linked list over array. *[(CO4)(Remember/LOCQ)]*
(c) Find the Big-Oh for $T(n) = 48n^{100} + 2^{n+2} + 3n^2 + 100$. *[(CO2)(Apply/IOCQ)]*
(6 + 1) + 3 + 2 = 12

Group - C

4. (a) Write a small C function (no need for main, header files etc.) to print the moves of the well-known Tower of Hanoi problem using recursion. Assume that the smallest disc is numbered as 'disc 1' and the largest is 'disc n'. Each of the three pegs is represented by a char (datatype): source ('S'), destination ('D') and temporary ('T'). Given below are some examples of the moves that you will print:
Move the disc # 12 from T to D
Move the disc # 4 from S to T. *[(CO3)(Understand/IOCQ)]*
(b) Now just write the output of your function if it is called with n = 4, where n is the number of discs. *[(CO3)(Apply/IOCQ)]*
(c) What will the following function return if it is called from the main as fun(5, 0, 1)? Can you identify what the following function is calculating? It is something well-known.

```
int fun(int n, int a, int b)
{
    if (n == 0)
        return a;
    if (n == 1)
        return b;
    return fun(n - 1, b, a + b);
}
```

[(CO3)(Analyse/IOCQ)]
5 + 2 + (4 + 1) = 12

5. (a) Why is it beneficial to use a circular array to implement a queue? Explain with an example. *[(CO4)(Remember/LOCQ)]*
(b) How is a circular linked list with a head pointer different from a circular linked list with a tail pointer? Which of the two would be better suited for implementing a queue and why, provided the list is allowed to have only one external pointer? *[(CO5)(Analyse/IOCQ)]*

(c) Stepwise show how the following infix expression is converted to its corresponding postfix expression.
((5+7)-(4*3))+(6/3)/2-9
Next, use a suitable data structure to evaluate the generated postfix expression. *[(CO3)(Apply/IOCQ)]*

(2 + 1) + (2.5 + 1.5) + (3 + 2) = 12

Group - D

6. (a) Define Binary Search Tree. *[(CO1)(Remember/LOCQ)]*
Consider a Binary Search Tree (BST) consisting of distinct integers lying in the range between 1 and 1000. In order to search (or probe) the number 363 in the given BST, which of the following search (or probe) sequences is correct one and which sequences are not correct. Justify your answer for each of these search (or probe) sequences.
(i) 2, 252, 401, 398, 330, 344, 397, 363
(ii) 924, 220, 911, 244, 898, 258, 362, 363
(iii) 925, 202, 911, 240, 912, 245, 363
(iv) 2, 399, 387, 219, 266, 382, 381, 278, 363
(v) 935, 278, 347, 621, 299, 392, 358, 363. *[(CO2, CO3, CO6)(Apply/LOCQ)]*

(b) The square of a directed graph G=(V,E) is the graph G2 =(V,E2) such that (u,w) in E2 if and only if for some v in V, both (u,v) in E and (v,w) in E. In other words, G2 contains an edge between u and w whenever G contains a path with exactly two edges between u and w. Describe the outline of an efficient algorithm for computing G2 from G for the adjacency-matrix representation of G. Analyze the running times of your algorithm. *[(CO2,CO3,CO4)(Apply/IOCQ)]*

(1 + 5) + 6 = 12

7. (a) (i) Consider a binary tree with **n** (i.e., **n >= 0**) nodes, which has been constructed using linked representation, where the structure of each tree node has three components: data, pointer to left subtree and pointer to right subtree. Suggest an algorithm (/ pseudo code / C-code) of a function (or procedure) to determine that whether the given binary tree is a Binary Search Tree (BST) or not.
[**N.B.** Assume that the function has access to the root node of the given binary tree.]
(ii) What is the best case time complexity of the above algorithm? *[(CO1,CO2,CO3,CO4)(Apply/IOCQ)]*

(b) Consider a search binary tree with **n** (i.e., **n >= 0**) nodes, which has been constructed using linked representation, where structure of each tree node has three components: data, pointer to left subtree and pointer to right subtree. Write a recursive algorithm that will display all the keys of a given Binary search tree in sorted (ascending) order. *[(CO1,CO2,CO3)(Apply/LOCQ)]*

(c) Given an undirected graph G=(V,E), let d(u, v) be the minimum number of edges on a path between u and v. Let the diameter of G be the maximum d(u, v) over all such pairs (u, v) in G. Explain how you may find the diameter of a graph using BFS. *[(CO2,CO3,CO6)(Apply/HOCQ)]*

(4 + 1) + 3 + 4 = 12

Group - E

8. (a) Show that the expected number of comparisons required to find an element from an array of n elements using sequential search, if the element occurs twice in the array, is (n + 1) / 3. *[(CO3)(Analyse/IOCQ)]*

(b) In the following pseudo-code for Heapsort, it is assumed that the array index starts from 1. Can you write the modified pseudo-code for these two procedures in case the array index starts from 0?

```
BUILD-MAX-HEAP(A)
1  heap-size[A] ← length[A]
2  for i ← ⌊length[A]/2⌋ downto 1
3       do MAX-HEAPIFY(A, i)
HEAPSORT(A)
1  BUILD-MAX-HEAP(A)
2  for i ← length[A] downto 2
3       do exchange A[1] ↔ A[i]
4          heap-size[A] ← heap-size[A] − 1
5          MAX-HEAPIFY(A, 1)
```

[(CO1,CO4)(Apply/IOCQ)]

(c) If a Heap is stored in an array starting with index = 1,
(i) What will be the index of the left child of a node whose index is 5?
(ii) What will be the index of the right child of a node with index 21?
(iii) What will be the index of the parent of a node, whose index is j? *[(CO1,CO2)(Apply/IOCQ)]*

5 + (2 + 2) + (1 + 1 + 1) = 12

9. (a) Write the non-recursive pseudo-code for Binary-search? State its best-case and average case complexity. *[(CO2)(Remember/IOCQ)]*

(b) Give an example of a situation where using counting sort is better than using any other comparison-sort technique? State the asymptotic time-complexity of counting sort, clearly defining the parameters that you use in your expression. *[(CO4)(Analyse/IOCQ)]*

(c) In a closed hashing we are using an array of size 10 whose 0, 1, 2, 5, 6, 9 indexed positions are already filled and the hash function is h(k) = (k + 10) %10. First draw the array in such condition with some probable values. Now calculate the probability, that the next key which is going to be inserted should **NOT** be placed at the position with index = 3.

[(CO4,CO5)(Apply/IOCQ)]

(4 + 1) + (1 + 1) + (2 + 3) = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	21.88	66.67	11.46