FUNDAMENTALS OF DATA STRUCTURE & ALGORITHMS
(INFO 2101)

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) What will be the output of the following code snippet for the list 1->2->3->4->5->6?

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

(a) 123456 (b) 135531 (c) 135135 (d) 135246

(ii) Given an expression ((A+B)*(C+(D*E))). Which data structure is applied to check for proper parenthesis?
(a) Stack (b) Queue (c) Tree (d) Linked List.

(iii) Which data structure is applied in Quick Sort Algorithm?
(a) Stack (b) Queue (c) Tree (d) Linked List.

(iv) The time complexity of Tower of Hanoi problem is
(a) $O(n^4)$ (b) $O(2^n)$ (c) O(nlogn) (d) $O(n^{\log n})$.

(v) When the array is already sorted and modified bubble sort is applied the time complexity is
(a) $O(n^2)$ (b) O(n) (c) O (log n) (d) O (n logn).

(vi) The prefix expression of the infix expression a * (b +c) / e – f
(a) /*a+bc-ef (b) -/*a+bcef (c) -/*+abcef (d) none

(vii) Consider a B+-tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node?
(a) 1 (b) 3 (c) 4 (d) 5.

(viii) In a circular queue with array length n, if front = (rear +1) % n then
(a) Queue is empty (b) Queue is overflow
(c) Queue has only one element (d) None of the above.

(ix) Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?
(a) 7 5 1 0 3 2 4 6 8 9 (b) 0 2 4 3 1 6 5 9 8 7
(c) 0 1 2 3 4 5 6 7 8 9 (d) 9 8 6 4 2 3 0 1 5 7.

(x) Preservation of functional dependency is ensured by which of the correctness rule of the fragmentation?
(a) Disjointness (b) Reconstruction
(c) Completeness (d) All of these.

Fill in the blanks with the correct word

(xi) The _____ sorting algorithm works in O(n) in the best case.

(xii) Deletion of an element from the end of a linked list requires the modification of _____ pointers.

(xiii) An algorithm runs in _____ time complexity if the following holds
0 <= C2g(n) <= f(n) <= C1g(n) for n >= n0.

(xiv) A threaded binary tree is a binary tree in which every node that does not have right child has a thread to its _____.

(xv) If n elements are sorted in a binary search tree, the asymptotic complexity to search a key in the tree will be _____.

Group - B

2. (a) Suppose that T1(n) and T2(n) are time complexities of two program fragments P1 and P2, where T1(n) = O(nlogn), and T2(n) = O(n). Derive the time complexity of P1 followed by P2. *[(CO3)(Analysis/HOCQ)]*

(b) Write an algorithm to find whether the data entered in a linked list creates a palindrome or not. *[(CO2)(Apply/IOCQ)]*

(c) Order the following functions based on their growth rate.
(i) $(3/2)^n$ (ii) n^2 (iii) nlogn (iv) (n+1)!
(v) $n^{3/2}$ (vi) 7/3n (vii) $2^{\log n}$ (viii) log log n. *[(CO3)(Analyze/HOCQ)]*

3 + 6 + 3 = 12

3. (a) Write an algorithm to add two matrices using their sparse format. Define required data structures beforehand. *[(CO2)(Apply/IOCQ)]*

(b) Write a recursive C function to reverse a singly linked list. Define required data structures for this purpose. *[(CO2)(Apply/IOCQ)]*

(c) Write an algorithm to delete the n^{th} node from a doubly linked list. Describe the required data structures beforehand. *[(CO2)(Apply/IOCQ)]*

(4 + 2) + 3 + 3 = 12

Group - C

4. (a) Justify for or against the statement:
"Tower of Hanoi problem runs in exponential time as the number of disc increases". *[(CO1)(Analyse/HOCQ)]*

(b) Write a C function to convert the infix to postfix expression. Use the same to convert the following infix to its corresponding postfix expression using stack. (Show all intermediate steps.)

A - B / C * (D + E – F) / G * H * J *[(CO2)(Apply/IOCQ)]*

(c) What is a priority queue? Which data structure is used to implement a priority queue? *[(CO1,CO2)(Understand/LOCQ)]*

3 + (5 + 2) + (1 + 1) = 12

5. (a) Construct the following Queue of characters where Queue is a circular array which is allocated eight memory cells.

QUEUE:

	9	1	5	7	3		

Describe the Queue (along with FRONT REAR values) as the following operations take place:

(i) What is the value of FRONT and REAR presently?
(ii) When *2* is added to the Queue?
(iii) Three times deletion done from the Queue.
(iv) *17, 18* are added to the Queue.
(v) Three times deletion done from the Queue.
(vi) *19* is added to the Queue.
(vii) Once deleted from the Queue.
(ix) Two times deletions done.
(x) Again deletion is performed. *[(CO2)(Apply/IOCQ)]*

(b) Write a C function to define the enqueue operation in a queue, following the given prototype.

void ENQUEUE (struct q *, int); *[(CO2)(Apply/IOCQ)]*

(c) How can we implement a queue using stack data structure? *[(CO2)(Apply/IOCQ)]*

5 + 4 + 3 = 12

Group - D

6. (a) Use the DFS algorithm to search the key value 5 from the given grpah in Fig.1. If we use BFS algorithm to search the same key value whether same/less/more number of steps are required? *[(CO4)(Apply/IOCQ)]*

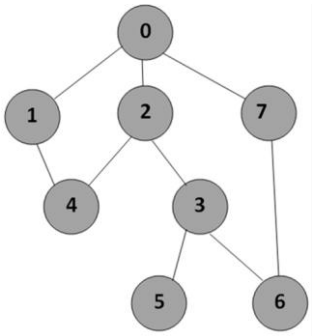

Fig.1

(b) "All circuits are closed walk but all closed walks are not circuits" — Justify. *[(CO4)(Evaluate/HOCQ)]*

(c) Find the shortest path using Dijkstra's shortest path Algorithm of the given weighted graph in Fig.2, (starting node 8). *[(CO4)(Apply/IOCQ)]*

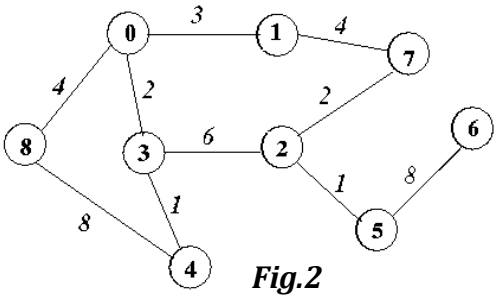

Fig.2

(d) Can there be multiple MST present for a given graph? Explain. *[(CO4)(Analyze/IOCQ)]*

4 + 2 + 4 + 2 = 12

7. (a) Proof that, if total N nodes are present in the binary tree, then in linked representation of it N+1 null links should be present. *[(CO2)(Analyse/IOCQ)]*

(b) Insert the following data into an AVL tree.
10, 20, 30, 40, 50, 25, 7, 9, 8
Clearly show every step with balancing. *[(CO2)(Apply/IOCQ)]*

(c) From the given traversals form corresponding binary tree: (Show each step carefully)
Preorder: A - B - D - I - J - F - C - G - K - H
Inorder: I - D - J - B - F - A - G - K - C - H *[(CO2)(Apply/IOCQ)]*

3 + 5 + 4 = 12

Group - E

8. (a) Write the binary search algorithm. Derive the time complexity of binary search with the help of a recursion tree. *[(CO3,CO5)(Apply/IOCQ)]*

(b) Transform the following array 12, 6, 7, 16, 19, 2, 11 into a max-heap using a suitable method and use the max-heap to sort the array. Show all intermediate steps. *[(CO5)(Apply/IOCQ)]*

(3 + 2) + 7 = 12

9. (a) Using divide-and-conquer approach, write the quick sort algorithm. *[(CO5)(Understand/LOCQ)]*

(b) "The best case, average case and worst case of quick sort algorithm are different from the other sorting algorithms" — Justify. *[(CO3)(Analyse/HOCQ)]*

(c) What is hashing. Explain linear probing and quadratic probing with example. *[(CO6)(Remember/LOCQ)]*

4 + 2 + 6 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	12.5	61.46	26.04

Course Outcome (CO):

After the completion of the course students will be able to

1. Develop the knowledge of basic data structures for storage and retrieval of ordered or unordered data.
2. Design linear and non-linear data structures to be used for storing, accessing and manipulating data,and be able to choose the appropriate data structure to be used for different real life applications.
3. Evaluate and compare the runtime and memory usage of algorithms with the help of mathematical background (Asymptotic Notation) of algorithm analysis.
4. Apply graph-based algorithms on shortest path problems.
5. Apply efficient algorithm for solving problems like sorting, searching, insertion and deletion of data.
6. Analyze hash functions and collision resolution techniques for storing and retrieving data efficiently into a hash table.

**LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question; HOCQ: Higher Order Cognitive Question.*