DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF BARISHAL
FINAL EXAMINATION
Course Title: Data Structures; Course Code: CSE-1201
1st Year 2nd Semester; Session: 2023-24

Time: 3 Hours **Marks:** 60

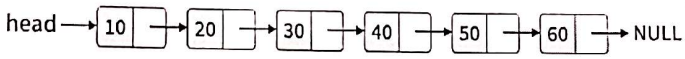
(Answer any FIVE questions from the followings. All parts of each question should be answered together.)

1. a) Define data structure. Why are data structures important in the development of efficient algorithms? Explain with suitable examples. [3]
   b) What is an abstract data type (ADT)? Explain linear and non-linear data structures with suitable examples. [3]
   c) Sorting is any process of arranging items systematically, and has two common, yet distinct meanings: ordering: arranging items in a sequence ordered by some criterion; categorizing: grouping items with similar properties. Now your task is to sort the elements using: [6]

      4, 10, 3, 5, 1, 15, 2, 4

      i. Insertion sort
      ii. Selection Sort

2. a) What is recursion? Write a recursive algorithm to sum natural numbers (1, 2, 3 ..., N). [4]
   b) Define a Linked List. Write a function to insert an element in the middle of a Linked List. Let consider the following linked list with 10 as the head. Now, write an algorithm to insert a new node (15) in the middle of node 30 and 40. [4]



   c) Describe a situation where storing items in an array is clearly better than storing items on a linked list. [2]

   d) An AVL tree stores the elements: < ***40, 20, 50, 10, 25, 30, 22, 27*** >. Show all rotations that occur during insertion. *[2]*

3. a) You are given an array of sorted roll numbers: **[10, 15, 22, 25, 34, 51, 73, 89, 95]** *[6]*

      i) Trace the complete steps of binary search for the key **73.**
      ii) Count the number of comparisons performed.
      iii) Generalize the worst-case time complexity of binary search and explain why it follows that pattern mathematically.

   b) [6]

      Merge sort is a classic recursive, divide-and-conquer sorting algorithm that divides an array into two halves, recursively sorts each half, and then merges the two sorted halves into a single sorted list. Now,

      i) Sort the following array using merge sort algorithm. Show all major recursive divisions and all merge operations step by step until the final sorted array is obtained.

      A = ⟨2, 8, 7, 1, 3, 5, 6, 4⟩

      ii) Some researchers propose using insertion sort for very small subarrays inside merge sort. Why insertion sort improves performance for subarrays below a certain threshold.

4. a) A hospital uses a **queue** to manage patient appointments. [4]
      i) Illustrate how patients are inserted and removed in a linear queue.
      ii) Explain what happens when the queue becomes full but has empty slots at the front (illustrate with an example). How does a circular queue solve this problem?

   b) The following sequence of stack operations is performed on an initially empty stack: [4]
      PUSH(10), PUSH(5), PUSH(3), POP(), PUSH(8), POP(), PUSH(12), PUSH(7), POP()
      i) Show the stack content after each operation.
      ii) Determine the total number of push and pop operations.
      iii) Explain how the LIFO property is reflected in the sequence above.

c) Consider the following deque of characters where DEQUE is a circular array which allocates six memory cells: [4]

LEFT = 2, RIGHT = 4, DEQUE: ------, A, C, D, ------, ------.

Describe the deque while the following operations take place:

i) F is added to the right of the deque.

ii) Two letters on the right are deleted.

iii) K, L, M are added to the left of the deque.

iv) One letter on the left is deleted.

v) R is added to the left of the deque.

5. a) Given a binary tree with **n** nodes. Derive the formula for the maximum number of nodes in a binary tree of height h. [2]

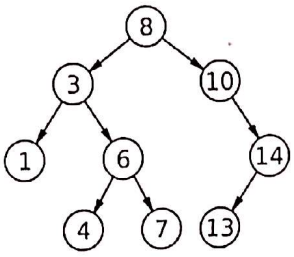
b) If a binary tree contains 40 nodes, determine the minimum and maximum possible height of the tree. Show the calculation steps. Explain how your results relate to balanced vs. skewed trees. [2]

c) Consider the following BST: [5]

i) Perform preorder, inorder, postorder, and level order traversals.

ii) If you insert a new node with value 65, where would it be placed in the BST formed by this tree? Show the mathematical comparison sequence.

iii) Show the trees after deleting the node 8.



d) Explain the term Balanced Binary Tree, Complete Binary Tree and Perfect Binary Tree with diagrams. [3]

6. a) Consider the array: **A = [4, 2, 2, 8, 3, 3, 1]** [6]

i) Simulate counting sort step by step, showing the count array, cumulative count array, and final sorted array. Explain why counting sort is stable.

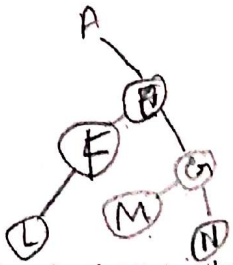
ii) Prove that the time complexity of counting sort is :$\Theta(n+k)$, where n=total number of elements, and k range of input keys. Explain why counting sort is not comparison-based and how this affects the lower bound of sorting algorithms.

b) Construct the binary tree from the following traversal sequences. Show all steps, including identification of the root at each stage and the subdivision of the left and right subtrees. [4]

Inorder: H D I B J E K A L F M C N G

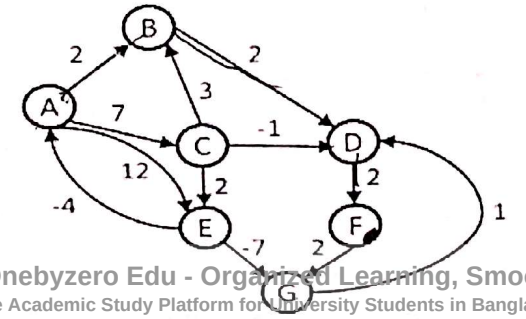
Postorder: H I D J K E B L M F N G C A

Draw the final binary tree and clearly mark left and right children.

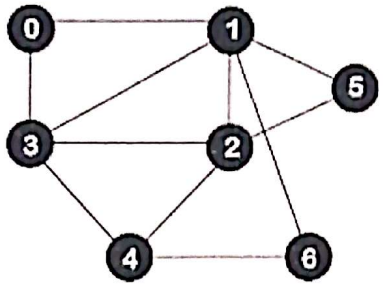


c) Differentiate between B tree and B+ tree. [2]

7. a) Consider the following directed, weighted graph. Now your task is to calculate the shortest paths from A to F. Show all possible paths also. [4]



b) Consider the following undirected graph: [6]



i) Perform a DFS traversal starting from node 1 and show the stack state at each step.

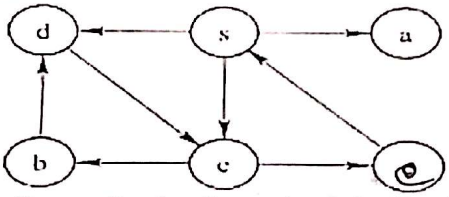
ii) Perform BFS traversal from node 1 and show the queue state at each step.

iii) Analyze which traversal discovers node 6 earlier and explain why.

c) Explain what is meant by biderctional and diagraph with examples. [2]

8. a) Define graph. Why linked representation of a graph is better than array representation? [3]

b) What are the advantages of adjacency matrix over adjacency list for graph traversal? Draw the adjacency Matrix and List for the below graph. [5]



c) Consider the following adjacent list for directed graph. Herein adj(y)=[x] refers to adjacent of y is x. [4]

adj(y) = [x],
adj(x) = [z],
adj(z) = [y,w],
adj(w) = [x],
adj(s) = [z,w],
adj(v) = [s,w],
adj(t) = [u,v],
adj(u) = [v].

i. Construct a graph from the above information.
ii. Identify the tree edge, back edge, cross edge and forward edge for the graph along with definition of those edges.

Good Luck!!!