Data Structures and Algorithms
Mid Semester Exam

25th February,2025

All questions are compulsory.

Question 1 (4+6 marks)

A. A sorting algorithm is said to be stable if two objects with equal keys appear in the same order in sorted output as they appear in the input unsorted array. Which of the below are not stable sorting algorithms and why?
i)Insertion sort ii)Merge Sort iii)Heap Sort iv)Quick Sort v)In-order traversal of AVL tree

B. An inversion in said to be present in an array A if A[i] > A[j] for i<j. What are the maximum number of inversions that can be present in an array A of unique elements. Present an algorithm (preferably O(n log n)) that finds all the inversions in the array.

Question 2 (3+7 marks)

A. Given an input sequence 1,2,3,4,5 which of the following permutations can be obtained using a stack
i) 3,4,5,1,2 ii)3,4,5,2,1 iii) 1,5,2,3,4. iv) 5,4,3,1,2

B. Given an array and an integer K, find the maximum sum you can make for each and every contiguous subarray of size K

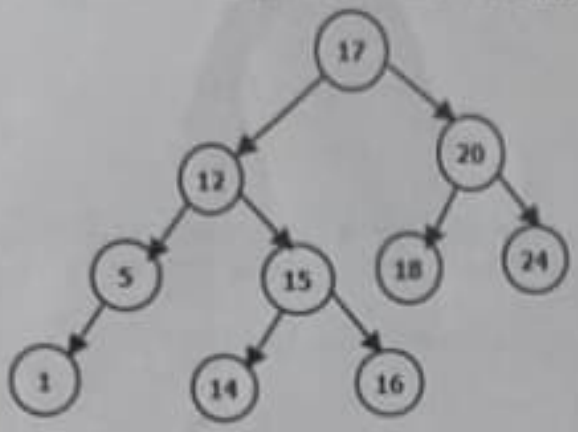
Question 3 (3+7 marks)

A. Consider a ternary heap where a node can have three children instead of two children. For an element x at index i in the corresponding array, what would be the index locations of the parent and children of the element x. [5 marks]

B. Given a range (k1,k2), how would you find the elements in a heap that lie within this range. That is report all values k where $k1 \leq k \leq k2$ and mention the complexity of doing so.

Question 4 (6 +4 marks)

A. Insert 13 into the AVL tree below and then delete 16 and then delete 15



depth

B. What is the maximum height difference possible between two leaf nodes in an AVL tree? Explain

2 0

Question 5 (4 +6 marks)

A. The in-order traversal of some binary tree produced the sequence HFIEJGZ, and the post-order traversal of the same tree produced the sequence HIFJZGE. What will be the total number of nodes in the left sub tree of the given tree?

B. Let E be an element present in a binary search tree as one of the leaf nodes. Let P be the set of elements present on the path that leads to E. Let L be the set of elements that do not belong to P and lie to the left of P. Let R be the set of elements not in P and lie to the right of P. In the example below, the nodes represented by ⊗ denote nodes in R while nodes marked as ◎ denote nodes in L. For any binary search tree, Is it always true that $l \leq p \leq r$ for any l in L,

p in P and r in R.?