This question paper consists of TWO (2) sections. Answer ALL questions in the answer booklet provided. [100 MARKS]

SECTION A (40 Marks)

There are TWENTY (20) questions in this section. Answer ALL questions in the answer booklet provided.

1. If h is any hashing function and is used to hash n keys in to a table of size m, where n<=m, the expected number of collisions involving a particular key x is :

   A. less than 1
   B. less than n
   C. less than m
   D. less than n/2

2. Let A be an adjacency matrix of a graph G. The ijth entry in the matrix A^K, gives

   A. The number of paths of length K from vertex Vi to vertex Vj
   B. Shortest path of K edges from vertex Vi to vertex Vj
   C. Length of a Eulerian path from vertex Vi to vertex Vj
   D. Length of a Hamiltonian cycle from vertex Vi to vertex Vj

3. Choose the correct statement regarding Selection Sort:

   A. Selection Sort has a time complexity of O(n log n).
   B. Selection Sort is an adaptive sorting algorithm.
   C. Selection Sort always performs the same number of comparisons regardless of the initial order of elements.
   D. Selection Sort is preferred for large datasets.

4. Which of the following statements about Quicksort is **TRUE?**

   A. Quicksort is a stable sorting algorithm.
   B. Quicksort has a time complexity of O(n log n) in the worst-case scenario.
   C. Quicksort uses the divide and conquer strategy.
   D. Quicksort is not suitable for sorting arrays with duplicate elements.

5. Which of the following sorting algorithms has the potential to have a worst-case time complexity of O(n log n) in practice but O(n^2) in the worst case?

   A. Selection Sort
   B. Quicksort
   C. Merge Sort
   D. Bubble Sort

6. Which sorting algorithm is more efficient for small datasets?

   A. Selection Sort
   B. Quicksort
   C. Merge Sort
   D. Bubble Sort

7. Quicksort uses a pivot element to partition the array. What is the role of the pivot?

   A. To compare with all elements in the array
   B. To divide the array into two smaller subarrays
   C. To move to the end of the array
   D. To perform a binary search

8. B Trees are generally ____.

   A. very deep and narrow
   B. very wide and shallow
   C. very deep and very wide
   D. cannot say

9. If a node having two children is deleted from a binary tree, it is replaced by its _____.

   A. in order predecessor
   B. in order successor
   C. preorder predecessor
   D. None of the above

10. A technique for direct search is __________.

   A. binary Search
   B. linear Search
   C. tree Search
   D. hashing

11. The searching technique that takes O (1) time to find data is ______.

   A. binary Search
   B. linear Search
   C. tree Search
   D. hashing

12. In the worst case Quick Sort has order ______.

A. O (n log n)
B. O (n^2/2)
C. O (log n)
D. O (n^2/4)

13. If a node in a Binary Search Tree has two children, then its predecessor has ______.

A. no left child
B. no right child
C. two children
D. no child

14. A binary tree in which if all its levels except possibly the last, have the maximum number of nodes and all the nodes at the last level appear as far left as possible, is known as ____.

A. full binary tree
B. AVL tree
C. threaded tree
D. complete binary tree

15. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a _____.

A. queue
B. stack
C. tree
D. linked list

16. The quick sort algorithm exploits _________ design technique.

A. Greedy
B. Dynamic programming
C. Divide and Conquer
D. Backtracking

17. The number of different directed trees with 3 nodes are _______.

A. 2
B. 3
C. 4
D. 5

18. A binary tree of depth "d" is an almost complete binary tree if _______

    A. Each leaf in the tree is either at level "d" or at level "d–1"
    B. For any node "n" in the tree with a right descendent at level "d" all the left descendants of "n" that are leaves, are also at level "d"
    C. Both (A) & (B)
    D. None of the above

19. What is the primary goal of maintaining AVL tree balance during insertion and deletion operations?

    A. To minimize the total number of nodes in the tree.
    B. To ensure that the tree height remains logarithmic, balancing the search and update time.
    C. To maximize the depth of the tree for efficient storage.
    D. To make the tree completely balanced at all times.

20. Which of the following AVL tree property is violated if the balance factor of a node is greater than 1 or less than -1?

    A. The Search Property
    B. The Heap Property
    C. The AVL Property
    D. The Symmetric Property

SECTION B **(60 Marks)**

There are FOUR (4) questions in this section. Answer ALL questions in an essay format.

Question 1 (10 marks)

Discuss the concept of a binary search tree (BST), highlighting its key characteristics, operations, and how it is used to efficiently organize and search data in various applications. Explore the advantages and limitations of BSTs compared to other data structures and provide examples of real-world scenarios where BSTs are commonly employed.

Question 2 (6 marks)

Discuss the concept of divide-and-conquer algorithms. How does this approach enhance the efficiency of algorithms? Provide examples of divide-and-conquer algorithms and explain their advantages.

Question 3 (24 marks)

Consider Selection-Sort and Merge-Sort. For each algorithm, what will be the worst case asymptotic upper bound on the running time if you know additionally that
(a) the input is already sorted.
(b) the input is reversely sorted.
(c) the input is a list containing n copies of the same number.

For each case and each sorting algorithm, state your answer and justify it in one sentence.

Question 4 (20 marks)

Imagine you are working for a transportation planning company responsible for optimizing routes in a city. Answer the following questions based on this scenario:

(a) How would you represent the road network as a graph data structure? What would be the nodes and edges in the graph?

(b) Suggest an algorithm to find the shortest path between two locations in the city, considering factors such as traffic conditions and road closures.

(c) Discuss the importance of edge weights in the context of transportation networks. How might different weightings affect the shortest path algorithm's output?

(d) How would you handle scenarios where the transportation network experiences changes, such as new roads being constructed, or existing roads being closed?

(e) Consider privacy concerns related to location data in transportation networks. How might you design algorithms or data structures to prioritize user privacy while still providing accurate route recommendations?

\

***** END OF QUESTION PAPER *****