**JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**SCHOOL OF INFORMATICS AND INNOVATIVE SYSTEMS**

**UNIVERSITY EXAMINATION FOR THE DEGREE OF BACHELOR OF ACTUARIAL SCIENCE WITH IT**

**3TH YEAR 2ND SEMESTER**

**2024/2025 ACADEMIC YEAR**

# MAIN CAMPUS

---

**COURSE CODE: ITB 9302 / SCS 318**

**COURSE TITLE: DESIGN AND ANALYSIS OF ALGORITHMS**

**DATE: 23/4/2025** **TIME: 15.00-17.00**

**TIME: 2 HOURS VENUE: …….**

---

## Instructions:

1. Answer Section A and any other TWO Sections
2. **Candidates are advised not to write on the question paper.**
3. **Candidates MUST hand in their answer booklets to the invigilator while in the examination room.**
4. **Mobile phones are NOT allowed in the examination room.**

## SECTION A: 20 Marks (Each question carries 1 mark)

1. Define an **algorithm**. (1 mark)
2. What is **asymptotic notation**, and why is it important in algorithm analysis? (3 marks)
3. Differentiate between **Big-O, Theta (Θ), and Omega (Ω)** notations. (3 marks)
4. State the **time complexity** of the following algorithms: (4 marks)
   a) Binary Search
   b) Merge Sort
   c) Quick Sort
   d) Dijkstra's Algorithm
5. Explain the **Divide and Conquer** strategy with an example. (3 marks)
6. How does **fractional knapsack differ from 0/1 knapsack**? (3 marks)
7. What is **Huffman coding**, and why is it used? (3 marks)
8. Compare **Prim's and Kruskal's Algorithm** for finding the minimum spanning tree. (3 marks)
9. What is the **Longest Common Subsequence (LCS)**, and where is it used? (3 marks)
10. Define **Backtracking** and give one example where it is used. (3 marks)



## SECTION B: 20 Marks

1. State Master's Theorem (5 marks)
2. Solve the following recurrence relations using **Master's Theorem**: (6 marks)
   $T(n) = 2T(n/2) + n$
   $T(n) = 4T(n/2) + n^2$
   $T(n) = 3T(n/3)+O(1)$
3. Solve the following recurrence relations using the **recursion tree method**: (4 marks)
   $T(n) = 3T(n/2) + n$
   $T(n) = 5T(n/4) + n^2$
4. Derive the time complexity of **Merge Sort** using **recurrence relations**. (5 marks)

## SECTION C:

1. How does **Branch and Bound improve the performance of the Traveling Salesman Problem (TSP)** compared to brute-force? (10 marks)
2. Critically analyze **why backtracking is inefficient** for large problem spaces and suggest alternatives. (10 marks)

## SECTION D: 20 Marks (10 marks each)

1. Describe the Breadth-First Search (BFS) algorithm for traversing a graph. Provide the key steps, pseudocode, and discuss its applications. Analyze the time complexity of BFS..
2. Define a binary search tree (BST). Describe the conditions that a binary tree must satisfy to be considered a BST. Provide an algorithm to find the minimum value in a BST.

## SECTION E: 20 Marks (10 marks each)

1. Explain the concept of a doubly linked list. Compare it with a singly linked list, highlighting advantages and disadvantages. Provide a code snippet for inserting a node at the end of a doubly linked list.
2. Compare and contrast QuickSort and MergeSort. Include their time complexity analyses, advantages, and situations where one might be preferred over the other. Provide pseudocode for both algorithms.