FINAL EXAMINATION-2024

Course Title: Design and Analysis of Algorithms

Course Code: CSE-2201

2nd Year 2nd Semester

Admission Session: 2022-23

Time: 3 hours

Answer any five Questions from the followings.

Marks: 60

1 a) Define an algorithm in your own words. Explain three characteristics of a good algorithm (clarity, efficiency, correctness) using a real-life example such as "ATM cash withdrawal process." [4]

b) A shopping app checks prices of n products one by one. Explain the meaning of O(n) time in this scenario with one simple example. [4]

c) Consider the following code: [4]

```
sum = 0
for i = 1 to n:
    sum = sum + 5
```

For n = 25, calculate the exact number of additions. Then state the time complexity in Big-O.

2 a) Explain Big-O, Big-Ω, Big-Θ using simple real-life analogies [4]

b) Show, using a simple inequality, that: [4]

$$6n^2 + 4n \leq 10n^2 \text{ for all } n \geq 1$$

Hence prove the function is $O(n^2)$.

c) A database search algorithm takes: [4]

- Algorithm A: 100n operations
- Algorithm B: $2n^2$ operations

For n = 20, calculate operations for A and B and determine which one is better.

3 a) Explain divide and conquer in real life using the example of organizing a college event (divide tasks → complete in groups → combine results). [4]

b) Sort the list step-by-step using divide and merge (merge sort): [50, 20, 40, 10, 30]. Show the following steps in details: [4]

- Split steps
- Merge steps
- Final sorted output

c) Quick Sort (first element as pivot): Perform one full Quick Sort procedure on: [32, 45, 12, 18, 5] and show the following steps: [4]

- Pivot selection
- Partition
- Recursive sublists
- Final sorted array

4 a) Explain how non-comparison sorting helps in real-life student grade processing (0–100) or sorting passport office token numbers. [4]

b) Use Counting Sort to sort the list: [4, 2, 2, 8, 3, 3, 1]. Show count array, cumulative frequency, and final sorted list. [4]

c) Consider Radix sort to sort the data: [329, 457, 657, 839, 436, 720, 355]. Explain why Radix Sort is used for fixed-length IDs in banks or warehousing. [4]

5 a) Explain MST using the real-life example of building a minimum-cost electricity network across villages. [2]

b) Consider Prim's algorithm and make a graph from the following vertex and edges: [5]

Edges: (A,B,4), (A,C,2), (B,C,1), (B,D,5), (C,E,10).

Build the MST and calculate total cost.

c) Show a small Bellman–Ford example with one negative edge and explain why Dijkstra fails. [5]

6 a) Why is Insertion Sort good for small or nearly sorted data in real-life situations like attendance list updates or class roll corrections? [2]

b) Apply Insertion Sort to the list: [5, 2, 4, 6, 1]. Write each pass clearly: [5]
- Which item you are inserting
- Comparisons made
- The list after each step

c) Apply Selection Sort to the list: [22, 11, 99, 88, 9] [5]
Show:
- The smallest element found in each pass
- The swap done in each pass
- The array after each swap

7 a) Explain the Naive String-Matching algorithm in simple terms. Use an easy example, such as searching the pattern "ANA" inside the text "BANANA". Describe how the algorithm checks each position, one by one, and what happens on mismatch. [4]

b) What is the prefix function / "failure table"? How it helps avoid repeating comparisons? Why KMP is faster for long texts. [4]

c) Compare Naive vs KMP string matching using the following points: [4]
- Number of comparisons (worst case)
- Time complexity (qualitative explanation)

8 a) Explain dynamic programming using a real-life travel optimization example. [4]

b) DP (Fibonacci): Fill the DP table for F(6). [4]

c) Explain NP-Complete problems using practical examples (exam scheduling, routing, timetabling). [4]