Complexity Theory & Mock Final Exam

DSA456 – Prepared by Azer Karadağ

Overview of Complexity Theory

- - Time & space complexity describe algorithm efficiency
- - Big-O: Worst-case time
- - Ω (Omega): Best-case time
- - Θ (Theta): Tight bound
- - P vs NP: Open question in CS
- - NP-Complete: Hardest in NP
- - Polynomial vs Exponential time

1. Time Complexity Classification [10 Marks]

- Classify the following algorithms by their time complexity:
- - Linear Search
- - Binary Search
- - Merge Sort
- - Bubble Sort (worst case)

- Answer:
- Linear Search: O(n)
- Binary Search: O(log n)

2. Pseudocode for Binary Search [10 Marks]

- Write pseudocode for the binary search algorithm on a sorted array.

- Answer:

```
function binarySearch(arr, target):
    left = 0
    right = length(arr) - 1
    while left <= right:
        mid = (left + right) // 2
```

3. Define NP-Complete [10 Marks]

- What does it mean for a problem to be NP-Complete?

- Answer:
- A problem is NP-Complete if:
- 1. It is in NP (verifiable in polynomial time)
- 2. Every other NP problem can be reduced to it in polynomial time.

4. Compare O(n) and O(log n) [5 Marks]

- Explain with an example why O(log n) algorithms are faster than O(n).

- Answer:
- Binary search (O(log n)) is faster than linear search (O(n)), because it halves the input size at each step.

5. Sorting Algorithm Time Complexities [10 Marks]

- List the best, average, and worst-case time complexities of Quick Sort.

- Answer:
- Best: O(n log n)
- Average: O(n log n)
- Worst: O(n^2)

6. Pseudocode for Factorial (Recursive) [10 Marks]

- Write a recursive pseudocode function to compute factorial of a number.

- Answer:

```
function factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

7. Identify Complexity from Code [10 Marks]

- Given the code:
- for i in range(n):
-     for j in range(n):
-         print(i, j)

- Identify the time complexity.

- Answer: O(n^2)

8. What is a Polynomial-Time Algorithm? [5 Marks]

- Give a definition and an example.

- Answer:
- Runs in O(n^k) time for some constant k.
- Example: Merge Sort runs in O(n log n).

9. Space Complexity Explanation [5 Marks]

- What is space complexity? Give an example of an algorithm with O(1) space complexity.

- Answer:
- Space complexity measures additional memory used.
- Example: Linear search (uses constant extra space).

10. Big-O, Big-Ω, and Big-Θ [5 Marks]

- Define Big-O, Big-Ω, and Big-Θ in your own words.

- Answer:
- Big-O: Upper bound (worst-case)
- Big-Ω: Lower bound (best-case)
- Big-Θ: Tight bound (average-case)