FINAL EXAMINATION

Specimen 2

Module: DSA-251: Data Structure & Algorithm

Time Allowed: 3 Hours

Total Marks: 100

Instructions:

Answer **all** four questions.

Each question is worth 25 marks.

Read each question carefully before answering.

QUESTION 1 [25 MARKS]

(a) Define Big O Notation and explain its primary purpose in the context of algorithm analysis.

[5]

(b) Explain the difference between solving a problem iteratively versus solving it recursively.

[5]

(c) In the context of C++ data structures using dynamic memory, explain what a memory leak is and how it typically occurs.

[5]

(d) Differentiate between an Abstract Data Type (ADT) and a concrete Data Structure.

[5]

(e) [Programming] Draw a flowchart for the following code:

```
int sum = 0;
for (int i = 0; i < 5; i++) {
    sum = sum + arr[i];
}
```

[5]

QUESTION 2 [25 MARKS]

(a) Describe the steps to insert a `newNode` after `targetNode` in a singly linked list.

[5]

(b) State the add/remove operations for a Stack and a Queue.

[5]

(c) What problem does a circular array-based queue solve compared to a normal array-based queue?

[5]

(d) Explain the role of the stack in checking balanced parentheses (example: `{[()]}`).

[5]

(e) Explain each of these lines in `pop()`:

```
int value = head->data;
Node* temp = head;
head = head->next;
```

[5]

QUESTION 3 [25 MARKS]

(a) Describe the recursive BST search algorithm.

[5]

(b) List two good properties of a hash function.

[5]

(c) Define a collision in hashing and name two resolution methods.

[5]

(d) Compare BST and Hash Table for dictionary use (average search complexities).

[5]

(e) Fill in missing lines in BST `insert`:

```
if (value < node->value) {
    node->left = insert(node->left, value);
} else if (value > node->value) {
    node->right = insert(node->right, value);
}
```

[5]

QUESTION 4 [25 MARKS]

(a) Name the two main graph representations and say which is better for sparse graphs.

[5]

(b) Describe DFS strategy.

[5]

(c) Describe BFS strategy.

[5]

(d) Difference between $O(n^2)$ and $O(n \log n)$ sorting with examples.

[5]

(e) C++ function to add directed edge to adjacency list.

[5]

END OF PAPER