UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 3, Examination in Engineering, February 2023

Module Number: EE3302 **Module Name: Data Structures and Algorithms**

[3 hours]

[Answer **all questions**, each question carries 10 marks]

Q1. a) An array is a linear data structure that stores data in contiguous memory locations.

i) State one (1) advantage and one (1) disadvantage of Arrays over other data structures.

[0.5 marks]

ii) Write a code to double the size of an array without losing the data.

[1.5 marks]

iii) State the two (2) types of memory used in storing data and provide a working example of how each memory is used.

[1.0 mark]

b) Linked data structures can store data in scattered memory locations.

i) State three (3) linked data structures.

[1.0 mark]

ii) You are asked to write a program to swap any two adjacent nodes in a singly linked list. Explain the steps with sequence numbers using a diagram.

[1.0 mark]

iii) Write a C++ function to implement the steps mentioned in Q1.b).ii). The function should take the first position of the two adjacent nodes as an input. Refer to Figure Q1.b.

[1.5 marks]

```
void swapNodes(int j) { //swap two adjacent nodes
    Node* curr = head;
    Node* prev = NULL;

    /*
    Write your code here
    */

}
```

Figure Q1.b: Code fragment.

c) A Doubly Linked List has links in both forward and backward directions.

i) Write a C++ program to insert a node at a given position in Doubly Linked List. You may call the "insertFirst" and "insertLast" functions to insert nodes at the first and last positions, respectively.

[2.0 marks]

ii) Compute the best, worst, and average case time complexities of the insert operation in part Q1.c).i) (use big O notation).

[1.5 marks]

Q2. a) Stack and Queue are two logical data structures.

i) State an application of Stack and Queue data structures each.

[0.5 marks]

ii) Using a working example, explain how to reverse a given Queue with the help of a Stack.

[1.0 mark]

iii) Write a C++ program to implement the process given in Q2.a).ii).

[1.5 marks]

iv) Explain how the Stack data structure can be implemented using Linked List.

[1.0 mark]

b) A Tree is a non-linear data structure, which can store values in scattered memory locations.

i) Identify the root node and leaf nodes of the tree given in Fig. Q2.b. What are the relationships between nodes A, B, D, and E?

[1.0 mark]

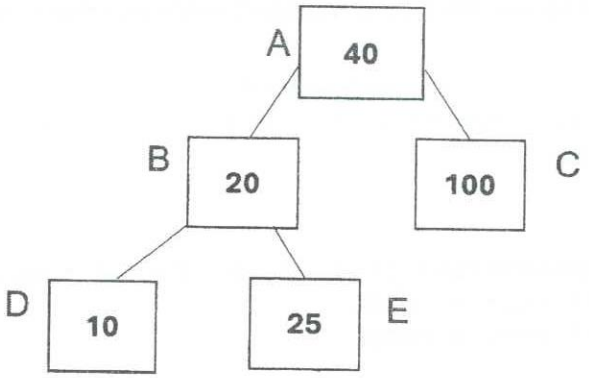

Figure Q2.b: Binary Tree

ii) Write a C++ program for a Node class in Binary Search Tree.

[1.0 mark]

iii) Compute the average time complexity of search operation in Binary Search tree.

[1.0 mark]

iv) Insert the following numbers in the given order to an empty Red-Black tree. You must draw the resulting tree after each insertion.

40, 60, 55, 30, 35, 25, 20, 10

[2.0 marks]

v) Write the order of the nodes printed in Post-order traversal, when applied on the Red-Black Tree created in part Q2.b).iv).

[1.0 mark]

Q3. a) Algorithm analysis is the process of evaluating the performance of an algorithm in terms of the amount of time and resources it requires to execute. Find the time and space complexity of the following code snippets with respect to input sizes N and M.

i)

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    a = a + i;
}
for (j = 0; j < M; j++) {
    b = b + i*i;
}
```

[1.0 mark]

ii)

```
int sum_array(int* A, int N) {
    int sum_n = 0;
    for (int i = 0; i < N; i++) {
        sum_n = sum_n + A[i];
    }
    return sum_n;
}
```

[1.0 mark]

iii)

```
int sum = 0
for( int i = N; i > 0; i /= 2 ) {
    for( int j = 1; j < N; j *= 2 ) {
        for( int k = 0; k < N; k += 2 ) {
            sum += k;
        }
    }
}
```

[1.0 mark]

iv)

```
void exchage_sort(int* A, int N) {
    for (int i = 0; i < N - 1; i++) {
        for (int j = i + 1; j < N; j++) {
            if (A[i] > A[j]) {
                //exchange A[i] with A[j]
                int temp = A[j];
                A[j] = A[i];
                A[i] = temp;
            }
        }
    }
}
```

[1.0 mark]

v)

```
int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int x = factorial(N)
```

[1.0 mark]

b) Hash table is based on the key-value lookup.

i) State two (2) applications of hashing.

[0.5 marks]

ii) Briefly explain the two (2) main collision handling techniques used in Hash tables.

[1.0 mark]

iii) Insert the entries in Table Q3.b in a Hash table of size 5. The Hash function will sum the value of each digit of the key and take the modules of the size of the table to generate the index. Show the workings and the final Hash table. Use linear probing as the collision resolution method.

[2.0 marks]

iv) What is the time complexity of insert operation without collisions in Hash table? Comment on how the time complexity changes when collisions exist if the chaining method is used to resolve them.

[1.5 marks]

Table Q3.b: Key-value pairs

Key	Value
0712391768	Nimal
0714567211	Kamal
0713831412	Senal
0712713723	Cyril

Q4. a) i) Explain the Insertion sort algorithm by using a diagram for a given array of elements. [2.0 marks]

ii) Give examples of input arrays for the worst case and best case running time for insertion sort. [1.0 mark]

iii) Calculate the best-case and worst-case asymptotic time complexity of the Insertion sort algorithm

[1.5 marks]

b) Merge sort algorithm is one of the fundamental sorting algorithms.

i) Write the function to merge two arrays of size N and return an array of size $2N$. [2.0 marks]

ii) Explain the Merge sort algorithm for the following array of elements, using diagrams for divide and merging steps.

```
34, 12, 2, 23, 7, 98, 5, 45, 9, 11, 34, 14, 3
```

[2.0 marks]

iii) Calculate the asymptotic time complexity of the Merge sort algorithm with the aid of the required diagrams.

[1.5 marks]

Q5. A graph is a data structure that is used to represent a set of objects (nodes or vertices) and the relationships between them (edges).

a) The graph shown in Figure Q5.a should be represented in a computer.

i) Using a diagram, show how the graph can be represented with an adjacency matrix.

[1.0 mark]

ii) Using a diagram, show how the graph can be represented with an adjacency list. [1.0 mark]

b) Consider the graph shown in Figure Q5.a.

i) Write the nodes in the order of traversal with Breath First Search Algorithm starting from node H0.

[1.5 marks]

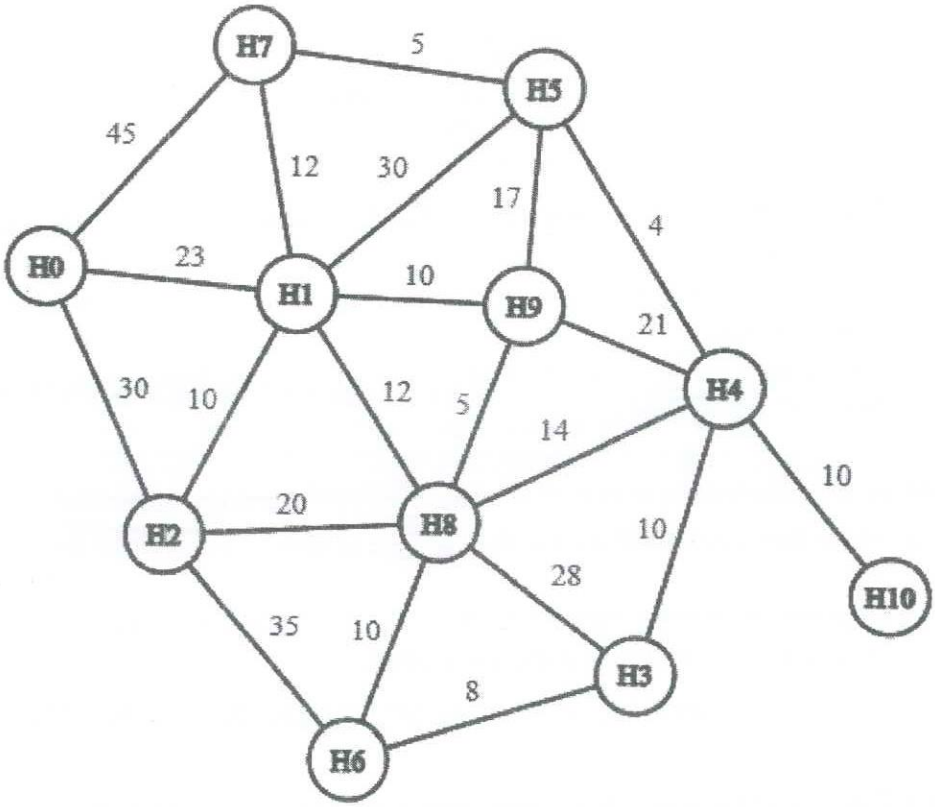

Figure Q5.a: A figure of an undirected graph.

ii) Write the nodes in the order of traversal with Depth First Search Algorithm starting from node H0. [1.5 marks]

c) You need to find the number of triangular paths in the Graph shown in Figure Q5.a. Write an algorithm where you can find the number of triangles using the Adjacency Matrix representation of the graph. [2.0 marks]

d) Finding the shortest path is a fundamental task in graph data structures.

i) Explain Dijkstra's algorithm for finding the shortest path. [1.5 marks]

ii) Using the graph illustrated in Figure Q5.a, show how Dijkstra's algorithm works for finding the shortest path starting from node H0. [1.5 marks]