



FINAL SEMESTER EXAMINATION

Programme	:	DIPLOMA IN COMPUTER SCIENCE (MQA/ PA13808)
Intake	:	JANUARY 2021
Course	:	DATA STRUCTURE AND ALGORITHMS
Course Code	:	DCS1353
Duration	:	(3 hours)

INSTRUCTIONS TO CANDIDATES:

- i. Please read the instructions given in the question paper **CAREFULLY**.
- ii. This question paper consists of **FOUR (4)** questions
- iii. Answer **ALL** questions in the question paper.
- iv. Answers to the questions are to be written into the examination booklet.
- v. Electronic dictionaries, lecture notes, files or any unauthorized materials except writing equipment are strictly prohibited.

This question paper must be submitted along with all used and/or unused rough papers and/ or graph papers (if any). Candidates are **NOT ALLOWED** to take any examination paper(s) used or unused out of the examination hall.

WARNING:

The Examination Board of Peninsula College Georgetown regards cheating as a very serious offence and will not hesitate to mete out the appropriate punitive actions according to the severity of the offence committed, and in the accordance with the clauses stipulated in the Students' Handbook, up to and including expulsion from Peninsula College Georgetown.

(This booklet contains 6 printed pages including this page)

For examiner's use only

QUESTION NO.	MARKS
1	/ 25
2	/ 25
3	/ 25
4	/ 25
Total	/ 100

Answer **ALL FOUR (4)** questions on the separate sheet provided.

[100 marks]

1. a) Describe the following worst-case running time of the following code in “big-Oh” notation in terms of the variable n . You should give the tightest bound possible.

i.

```
void f1(int n) {
    for(int i=0; i < n; i++) {
        for(int j=0; j < n; j++) {
            for(int k=0; k < n; k++) {
                for(int m=0; m < n; m++) {
                    System.out.println("!");
                }
            }
        }
    }
}
```

 (3 marks)

ii.

```
void f2(int n) {
    for(int i=0; i < n; i++) {
        for(int j=0; j < 10; j++) {
            for(int k=0; k < n; k++) {
                for(int m=0; m < 10; m++) {
                    System.out.println("!");
                }
            }
        }
    }
}
```

 (3 marks)

iii.

```
int f3(int n) {
    int sum = 73;
    for(int i=0; i < n; i++) {
        for(int j=i; j >= 5; j--) {
            sum--;
        }
    }
    return sum;
}
```

 (3 marks)

iv.

```
int f4(int n) {
    if (n < 10) {
        System.out.println("!");
        return n+3;
    } else {
        return f5(n-1) + 1;
    }
}
```

 (3 marks)

- b) List **THREE (3)** types of sorting available in java. (3 marks)

- c) Differentiate between iterative and recursive approach. (5 marks)

- d) State either **True** or **False** for the following statements. No explanation necessary.
- i. HeapSort has worst-case time complexity of $O(n \log(n))$. (1 mark)
 - ii. HeapSort makes no more than $O(n^2)$ pairwise comparisons. (1 mark)
 - iii. MergeSort has best-case time complexity of $O(n)$. (1 mark)
 - iv. InsertionSort make no more than $O(n \log(n))$ pairwise comparisons. (1 mark)
 - v. SelectionSort is stable. (1 mark)
- Total: [25 marks]

2.

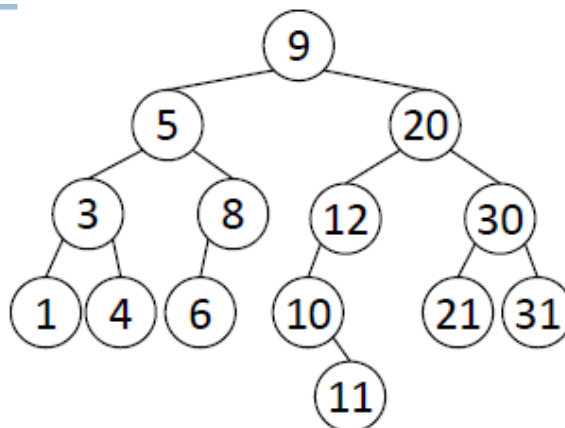


Figure 1

- a) Give the preorder, in-order, and post-order traversal based on Figure 1 (6 marks)
 tree.

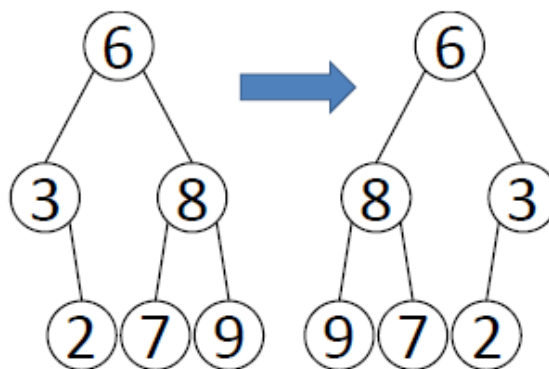


Figure 2

- b) Construct a simple Java program use a recursive method `public static Node mirror (Node node)` that changes a given input tree so that it becomes the mirror image of the original tree. (8 marks)
 (Note: Assume you already have the following code. Assume the method bodies, even though not shown, are correct and implement the operations as we defined them in class.)

```
public static void main (String args {...})
public class MirrorTree {...}
public Static Class Node {...}
public static class BinaryTree {...}
public static NodeMirror (Node node) {...}
void inorder {...}
```

- c) Construct a Java program for simple operations on binary search trees (11 marks) where add a method `int maxDepth ()` to the public static class `BinaryTree ()` class that computes the height of the tree. You will need a helper method.

(Note: Assume you already have the following code. Assume the method bodies, even though not shown, are correct and implement the operations as we defined them in class.)

```
public class BinarySearchTreeNode {...}
public static class node {...}
public static class BinaryTree{
int maxDept(Node node)}
public static void main (String args {...})
```

Total: [25 marks]

3.

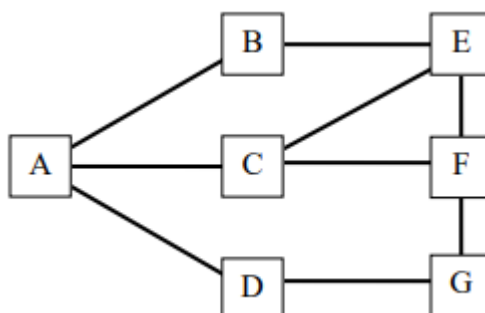


Figure 4

Figure 4 shows a computer network.

- a) Define the following terms
- i. Graph (2 marks)
 - ii. Undirected Graph (2 marks)
 - iii. Directed Graph (2 marks)
 - iv. Edge Attributes (2 marks)
- b) State the type of graph show in Figure 4 with **TWO (2)** further examples (2 marks) of graph applications with the following criteria.
- i. an undirected graph whose edge attributes are distances (2 marks)
 - ii. a directed graph whose edge attributes are flow rates. (2 marks)

- c) Suppose that a message is to be sent through the computer network of Figure 4, from A to G. It is possible that some computers are down at the time. The message must be sent through as few computers as possible but must not be sent through a computer that is down. State is the shortest route for the following message:
- i. when no computers are down; (2 marks)
 - ii. when computers B and D are down; (2 marks)
 - iii. when computers B, D, and F are down? (2 marks)
- d) Describe a general graph algorithm that can be adapted to solve problem based on Figure 4. (5 marks)
(Note: You may assume that any computer can be asked whether it is up or down. You may also assume that both A and G are up.)
- Total: [25 marks]
4. a) Explain the following algorithms.
- i. Greedy (4 marks)
 - ii. Dijkstra's Algorithm (4 marks)
 - iii. Divide and conquer (4 marks)
- b) Construct an algorithm for Dijkstra's Shortest Paths Algorithm with the following pre and post conditions. (10 marks)
- (Precondition: $G = (V, w)$ is a weighted graph with initial vertex v_0 .)
(Postcondition: Each vertex v in V stores the shortest distance from v_0 to v and a back reference to the preceding vertex along that shortest path.)
- c) State **TWO (2)** examples of Dijkstra's algorithm applications. (3 marks)
- Total: [25 marks]

- END OF QUESTIONS -