ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **WINTER SEMESTER, 2019-2020**
DURATION: 1 Hour 30 Minutes **FULL MARKS: 75**

CSE 4303: Data structures

Programmable calculators are not allowed. Do not write anything on the question paper.
There are **4 (four)** questions. **Question no 4 is Mandatory to answer.**
Answer any **2 (two)** from the remaining.
Figures in the right margin indicate marks.

---

1. a) Suppose a C++ parser is created which uses Reverse Polish notation. The following statement exhibits the result after being parsed using this parser. 7

```
a value . 3 4 s sin * + =
```

Here '*sin*' is a function of one argument and '.' is the member access operator, while 'a' and 's' are variables. How would this statement appear in the normal C++ programming language?

b) Implement a Queue using a Singly Linked List. The operations En-queue and De-queue should take $O(1)$ time. Explain the usefulness of Circular Queue in this scenario. 12

c) Find the Time-complexity of the following program assuming n is the size of the input data and b is a positive integer greater than 1. 6

```
1. void main(){
2.    j=1;
3.    Repeat step 4 and 5 while j<=n
4.      Function_1();
5.      j=b×j;
6. }

1. void Function_1(){
2.    i=2;
3.    repeat step 4 while i<=1000
4.      i=i×i;
5.    cout<<i;
6. }
```

2. a) A tree has Pre-order-depth-first and Level-order traversals respectively defined as: 6

```
A B C D E G F
A B C D E F G
```

Determine the original tree from this information.

b) Briefly propose an algorithm to implement a First-in-First-out queue with a priority queue. 7

c) With proper mathematical arguments prove that, the process of building a Max-heap from an arbitrary set of numbers can be done in linear time. 12

3. a) Perform the following operations in an AVL Tree: 15

```
insert(100)
insert(150)
insert(200)
insert(250)
insert(225)
insert(210)
insert(110)
delete(150)
insert(220)
insert(205)
insert(115)
delete(110)
delete(100)
delete(210)
```

Design a sorting algorithm which will show all the elements of this tree in descending order. What will be the time complexity of the process of sorting?

b) Show that, "If a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child". 10
Justify the statement again considering the condition of having two children being withdrawn.

[Mandatory]

4. a) The n^{th} Fibonacci number is defined as the sum of the two previous Fibonacci numbers where the 0^{th} and 1^{st} Fibonacci numbers are defined as 1. Suppose that each all of the function occupies 100 bytes on the call-stack. 10

```
int Fibonacci(int n){
    return (n<=1) ? 1 : Fibonacci(n-1) + Fibonacci(n-2);
}
```

What is the maximum size of the stack when this function is called with the argument n = 4? What is the total number of function calls made, including the initial call? Draw necessary figures to justify your answer.

b) What is the difference between the Binary-search-tree property and the Min-heap property? Can the min-heap property be used to print out the keys of an n-node tree in sorted order in $O(n)$ time? Justify your answer. 8

c) Where can the smallest element reside in a max-heap, assuming that all the elements are distinct? What will be the worst-case time complexity to find that element? (use tight-bound) 7