00426

The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical and Computer Engineering

Study Programme	: Bachelor of Software Engineering Honours
Name of the Examination	: Final Examination
Course Code and Title	: **EEX4465 Data Structures and Algorithms**
Academic Year	: 2022/23
Date	: 28th May 2023
Time	: 13:30 – 16:30 hrs.
Duration	: **3 hours**

General Instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **seven (7)** questions in **four (4)** pages.
3. Answer any **FIVE (5)** questions only. All questions carry equal marks.
4. Answer for each question should commence from a new page.
5. Answers should be in clear handwriting and do not use Red colour pen.
6. Clearly state your assumptions, if any.
7. This is a **Closed Book Test** (CBT).

Q1

(a) Briefly explain why the *Time complexity* is significant in algorithms. [02]

(b) A quadratic algorithm with processing time $T(n) = c\ n^2$ spends $T(N)$ seconds for processing N items, and c is the constant. How much time will be spent for processing $n = 5000$ items, when $N = 100$ and $T(N) = 1$ ms? [03]

(c) The following expressions give the processing time spent by an algorithm for solving a problem of size n. By explaining your work find the worst-case complexity (Big-Oh) of each problem. [15]

i)

```
public static int A(int n) {
    if (n == 1) return 0;
    return 1 + A(n/2);
}
```

ii)

```
public static int B(int n) {
    if (n == 0) return 1;
    return B(n - 1) + B(n - 1);
}
```

iii)

```
public static int C(int n) {
    if (n == 0) return 0;
    return C(n/2) + E(n) + C(n/2);
}
```

iv)

```
public static int D(int n) {
    if (n == 0) return 1;
    int x = 0;
    for (int i = 0; i < n; i++)
        x += D(n - 1);
    return x;
}
```

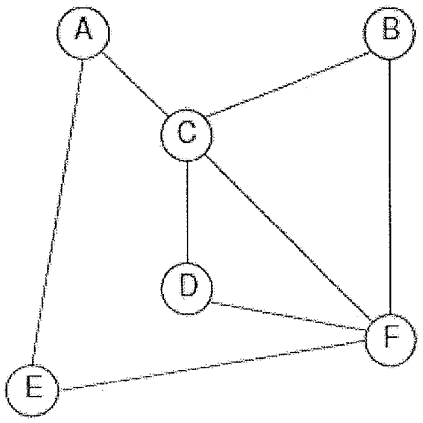
v)

```
public static int E(int n) {
    int x = 0;
    for (int i = 0; i < n; i++)
        x++;
    return x;
}
```

Q2

(a) Write the pseudocode algorithm to apply the breadth-first search, starting at any vertex for a given directed graph. [06]

(b) Apply the algorithm written in (a) to obtain the spanning tree of the following undirected graph, starting at vertex **A**. Show all steps with the data structure you used. [14]



Q3

(a) Consider the infix notation given by

$$3 \times (1 + 8/2)\text{^}3$$

i) Write the above expression into its equivalent postfix notation. [01]

ii) Evaluate the postfix notation obtained in i) by showing the contents of the stack at every step, by creating the table with the column headers as shown below. [05]

Next symbol	Operand 1	Operand 2	Value	Stack
:	:	:	:	:
:	:	:	:	:

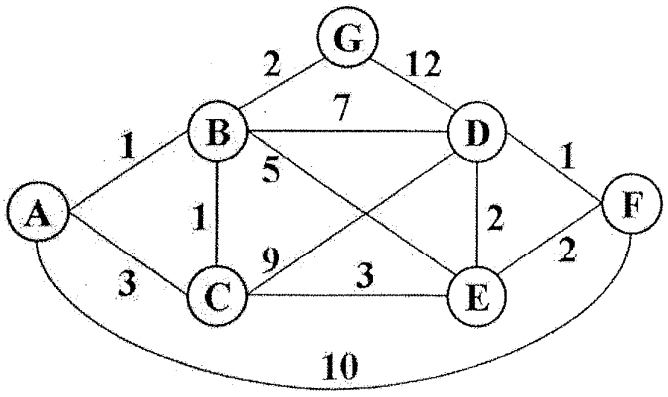
(b) Convert the following algebraic expression to *postfix* form using a stack as in the table below. Show the steps for each character: [14]

$$A/(B + C)\text{^}(D - E)$$

Reading character	Postfix	Stack

Q4

Consider the following undirected, weighted graph to answer the questions below.



(a) Step through Dijkstra's algorithm to calculate supposedly shortest path from **A** to every other vertex. Show your steps in the table form. Cross out old values and write in new ones, from left to right, as the algorithm proceeds. [18]

(b) Obtain an adjacency matrix for the above graph. [02]

Q5

(a) Write complete pseudocode algorithms for the below operations in Binary Search Tree (BST):

i) To create a node and obtain pre-order traversal of a BST using recursive function. [07]

ii) To insert a given value into a constructed BST using recursive function. [04]

(b) Consider the sequence of numbers given below.

{50, 10, 40, 25, 75, 70, 35, 37, 90, 60}

i) Construct the BST for the above sequence by taking the first element as the root. [05]

ii) Obtain the pre-order traversal of the BST drawn in i). [02]

iii) Re-draw the BST after inserting the new element 65 to the tree drawn in i). [02]

Q6

(a) Give one advantage and one disadvantage of the linear probing technique. [02]

(b) Insert the objects with key values {15, 17, 8, 23, 3, 5}, respectively, into an initially empty hash table of size 7 with the hash function $h(x) = x$ MOD 7, by using the following methods. Show the resulting table for each method.

i) Open addressing with linear probing; [03]

ii) Closed addressing with direct chaining; [03]

(c) Consider an implementation of a hash table with linear probing as given below:

> It uses an array, "***storage***" to keep pointers to the keys, and uses a Boolean array, "***used***" to record which locations in "***storage***" have contained a "***key***".

Write the function, "***lookup***(. . .)" in pseudocode for this hash table implementation, searching for a given "***key***", and returning (as a Boolean) whether it was found. [12]

Q7 Merge sort follows the rule of *Divide and Conquer* to sort a given set of numbers/elements, recursively. Answer to the followings:

(a) Briefly describe the rule of *Divide and Conquer*. [04]

(b) What are the worst-case time complexity and space complexity of Merge sort. [02]

(c) Write the pseudocode algorithm for the Merge sort [08]

(d) Use the Merge sort algorithm to sort the following sequence of numbers. Show the complete step-by-step procedure [06]

{14, 7, 3, 12, 9, 11, 6, 12}

-- End --