THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF COMPUTER SCIENCE
B.Sc. DEGREE PROGRAMME : LEVEL 03
CPU1142- DATA STRUCTURES & ALGORITHMS
FINAL EXAMINATION – 2016/2017

DURATION:Two Hours (2 Hours)

Date: 09.01.2018 **Time: 1.30p.m. – 3.30 p.m.**

Answer FOUR (4) Questions ONLY.

QUESTION 01

1) What are the two main types of recursions that can be used with functions? Explain each type with an example.

2) Write codes in C programming language to do the following.
   a. Create a data structure to implement a **stack**. The structure should contain a variable to store the **top** position of the **stack** and an **array** to hold numeric data of type **int** in the **stack**. Declare a stack with the name "**mystack**".
   b. Write a function in C to **insert** a data item into "**mystack**". Use **void StackInsert(int x)** as the function header (check stack overflow condition before you insert a data item).
   c. Write a function in C to **delete** a data item from "**mystack**". Use **int StackDelete ()** as the function header. (Check the stack underflow condition before you delete a data item).

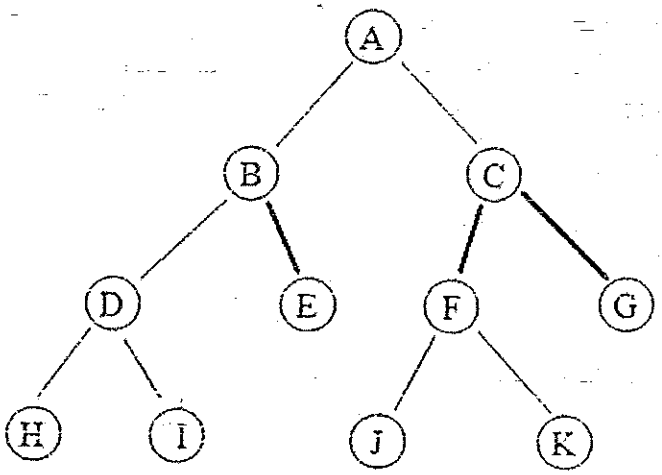
3) Using **big O** notation, determine the running time of the following C program section. State any assumptions you make.

```
for(i=1;i<n;i++)
  for(j=0;j<i;j++) printf("(%d,%d)\n",i,j);
```

4) Clearly describe the tasks of the following C program section.

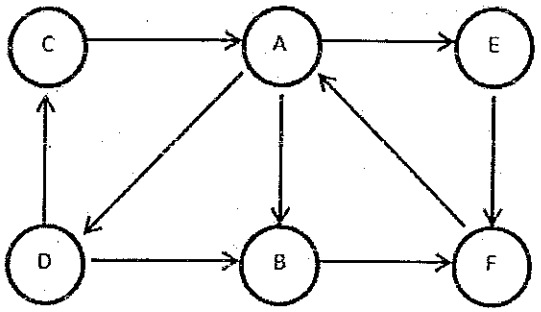
```
ptr = (int*) malloc(sizeof(int));
```

5) Define an "**almost complete binary tree**". Is the following tree an almost complete binary tree? Give reasons.



QUESTION 02

1) Show the **adjacency-list** representation of the following graph.



2)

a) Construct a binary search tree for the following set of integers.

60, 20, 100, 150, 80, 10, 50, 55, 05, 40, 90, 15, 70

b) What will be the output when you traverse the above binary tree, which you constructed in part (a) above in the following orders?

i. Pre-order
ii. In-order
iii. Post-order

c) Is the binary search tree you constructed in part (a) above a strictly binary tree?

d) What are the **leaf** nodes of the binary search tree which you constructed in part (a) above?

3) Describe the following terms with respect to **GRAPH** data structure.

   a) Degree of the vertex
   b) Complete Graph
   c) Weighted Graph
   d) Directed Graph

4) State the three types of internal sorting algorithms.

QUESTION 03

1) Write a function in C programming language to implement the **Selection Sort**. Use **void SelectionSort (int numbers[], int array_size)** where the array named "**numbers[]**" has the elements to be sorted and "**array_size**" gives the number of elements.

2) What is the running time of the selection sort (use **big O** notation)?

3) What will be the output of the following function if we pass "**5**" as the value for **n**?

```
int hanoi(int n)
{
        if (n == 1)
                return 1;
        else
                return 2*hanoi(n-1) + 1;
}
```

4) Write codes in C programming language to do the following.

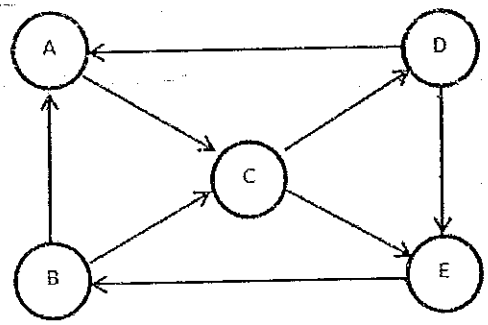
   a) Using **pointer implementation** of a queue, create a data structure of a queue.
   b) **Insert** an element to the **queue**.

QUESTION 04

1) Represent the following expression by using a binary tree.

   (A+B*C)$((D+E)*F)

2) Show the **multi-list representation** of the following graph.



3) Using the graph of part (2) above, show the order of vertices visited in the "**Depth first**" and "**Breath first**" traversals. Select "**B**" as the starting node.

4) Graphically show the steps of sorting the following data set by using the **Bubble sort**.

35	67	58	47	22	102	96	43

QUESTION 05

1) What is **Greedy Method** and give one example for that?

2) Write codes in C programming language to do the following.

   a) Using **pointer implementation**, create a data structure of a **circular queue**.
   b) **Insert** an element to the **circular queue**.

3) Consider the following expression in **Infix** form and convert it into the **Postfix** form. Clearly show the 7 **steps** required for the conversion.

(A+B)/C$D-E/F*G$H

4) Following is a function in C programming language for some operation.

```
void functionname()
{
        nodePtr *current, *first;
        current = last;

        if(current == NULL){
                return;
        }
        else if(current == current->next){
                free(current);
                last = NULL;
        }

        else{
                first = current->next;
                current->next = fist->next;
                free(first);
        }
}
```

By analyzing the above function in C programming language, answer the following questions.

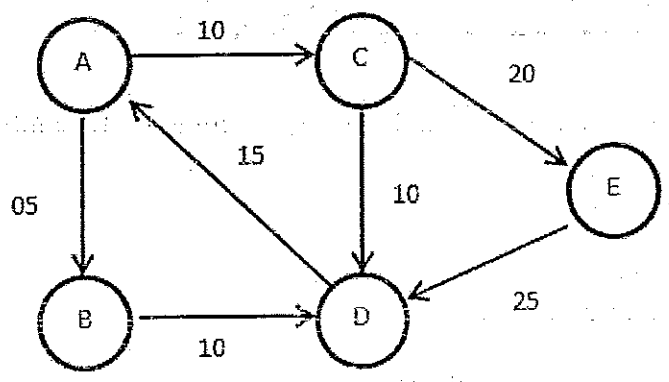
a) What is the data structure that this function belongs to?
b) What type of implementation method is used to create the data structure that you mentioned in section (a)?
c) Which type of operation can perform using the above function?

QUESTION 06

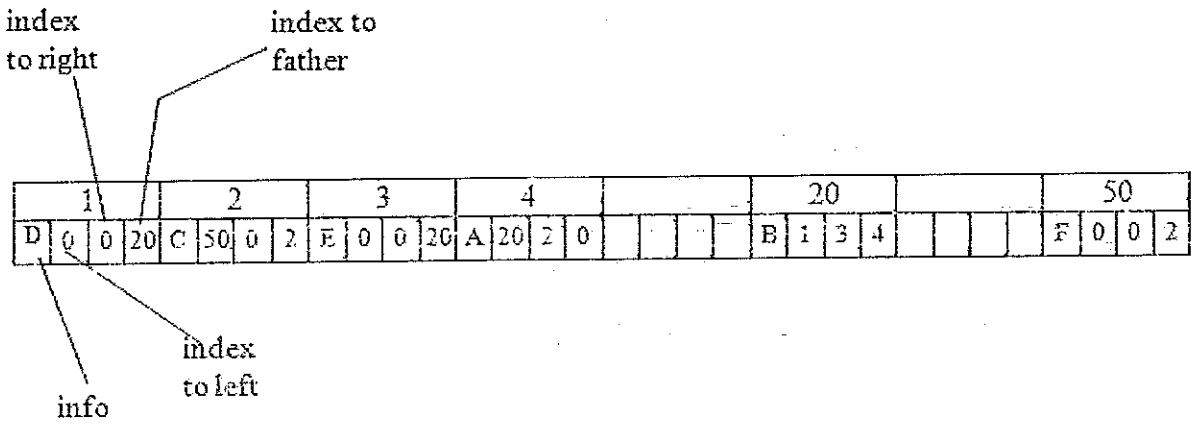
1) Using **big O** notation, determine the running time of the following **C** program section. State any assumptions you make.

```
for(i=0;i<n;i++) printf("%d\n",i);
for(j=0;j<n;j++) printf("%d\n",j);
```

2) Show the **adjacency-list** representation of the following graph.



3) Array representation of a binary tree is as follows.



a) Draw the relevant binary tree according to the above array representation.
b) Mention the degree of each node of the binary tree which you constructed in part (a) above?

4) Consider the running times **O(f(n))** and **O(g(n))** where

$$f(n) = n^3 \text{ if } n \text{ is even}$$
$$n^4 \text{ if } n \text{ is odd}$$

$$g(n) = n^6 \text{ if } n \text{ is even}$$
$$n^8 \text{ if } n \text{ is odd}$$

Consider that **f(n)** and **g(n)** are placed sequentially.

a) Calculate the time complexity for both **odd n** and **even n** separately.
b) What is the name of the **rule** that is required to calculate the time complexity?