



UNIVERSITY OF GHANA

BA / BSC FIRST SEMESTER UNIVERSITY EXAMINATIONS:2015/2016
COMPUTER SCIENCE
CSIT313 – PROGRAM DESIGN AND DATA STRUCTURES (3 CREDITS)

INSTRUCTIONS:

The question paper consists of Two sections, Section A and Section B.

In Section A, attempt All questions.

In Section B, answer any Two questions.

Answer Booklets will be provided.

TIME ALLOWED :

TWO AND HALF (2 ½) HOURS

Section A (50 MARKS)

- A1. Define Data Structure. [2 marks]
- A2. Define Abstract Data Type. [2 marks]
- A3. Define Record. [2 marks]
- A4. Define Linked List. [2 marks]
- A5. Define Tree. [2 marks]
- A6. Illustrate Binary Search Tree (BST) [2 marks]
- A7. Describe the operating circumstances under which a programmer will implement a linked list instead of a contiguous list. [3 marks]
- A8. What are the characteristics of data on which sequential search is best implemented? [3 marks]
- A9. Copy the table below and fill in for the times to sort an array of **n** elements. Use only **big-O notation**, and do not have any extraneous constants in your expressions. [4 marks]

Sorting Algorithm	Worst Case scenario	Best Case scenario
Bubble sort		
Selection sort		
Quicksort sort		
Mergesort		

- A10. List any six factors a programmer must consider in choosing a sorting method. [3 marks]
- A11. Give one practical example of the implementation of priority queues in computer systems.

[1 mark]

A12. Which data structure provides an excellent implementation of priority queues. [1 mark]

A13. List any two advantages of using array data structure in processing. [2 marks]

A14. Give three operating conditions under which binary trees are especially appropriate. [3 marks]

A15. Name the appropriate data structure for implementing :-

a. Breadth-first traversal of a graph. [1 mark]

b. Depth-first traversal of a graph. [1 mark]

A16. Give two advantages of an ADT. [2 marks]

A17. The code below is the content of a file “**queue.h**” that gives the declaration of a class **queue**. The class is meant to manage a queue of integers pointed to by “element” which is dynamically allocated. [14 marks]

```
#ifndef __QUEUE_H__
#define __QUEUE_H__

class queue
{
private :
    int size ;
    int * element ;
    int head ;
    int tail ;
public :
    queue(int s) ;
    void enqueue(int val) ;
    int dequeue() ;
    bool isFull() ;
    bool isEmpty() ;
    int front() ;
    int back() ;
    ~queue() ;
};
#endif
```

Required:-

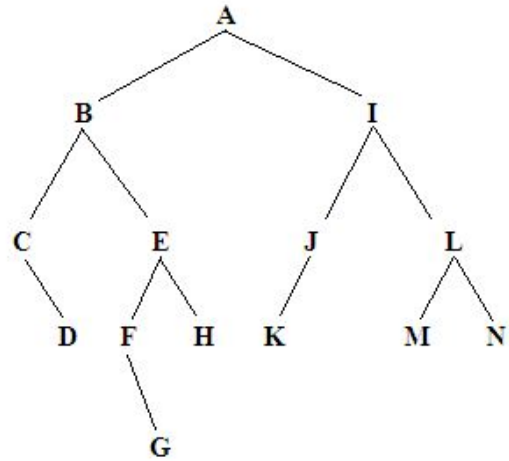
Write the content of file “**queue.cpp**” that is meant to *define all the member functions* in class **queue**.

Section B (50 MARKS)

In this section, answer **any two** questions. Each question carries an equal mark of 25.

B1. [25 marks]

a. Presented below is a Tree Structure.



i. What's the result of "Preorder Traversal"?

II. What's the result of "Inorder Traversal"?

II. What's the result of "Postorder Traversal"?

II. Is the above a binary search tree? Give reasons for your answer.

a. With the aid of a stack (represented as diagram(s)), give a stepwise evaluation of the reverse polish notation expression

12 8 7 - 1 + * 4 6 + 2 / -

a. Consider the searching problem:

Input: A sequence of **n** numbers **A** = **< a₁ , a₂ , ..., a_n >** and a value **v**.

Output: An index **i** such that **v = A[i]** or the special value **-1** if **v** does not appear in **A**.

Required:

Write pseudo code for linear search, which scans through the sequence **A**, looking

for v.

B2.

[25 marks]

a. Consider the graph below.

Represent the graph using

I. Adjacency list representation

I. Adjacency matrix representation

a. A singly linked list contains nodes which, in addition to a pointer storing the address of the next node and data as indicated in the class declaration (contained a file “link.h”) below. Insertion occurs at front.

```
#ifndef __SLINK_H__
#define __SLINK_H__
#include<cstdlib>
class Node
{
public :
    int info ;
    Node * next ;
    Node(int val)
    {
        info = val ;
        next = NULL ;
    }
} ;
```

```

class Slink
{
    private :
        Node * head ;
    public :
        ~Slink() ;
        Slink() ;

        void add(int item) ;
        bool find(int item ) ;
        int first() ;
        int removehead() ;
        int removelast() ;
        int remove(int item);
        void printAll() ;
        bool isEmpty() ;
} ;
#endif

```

Required :-

Implement all the member functions including the constructor and destructor.

a. Give a diagrammatic representation of how the list will appear if the following elements are added to the list in the order in which they occur 6 , 8 , 9 , 1 , 4 , 5.

B3. [25 marks]

a. Consider the following pseudocode:

```

declare a stack of characters
while ( there are more characters in the word to read )
{
    read a character
    push the character on the stack
}

while ( the stack is not empty )
{
    pop a character off the stack
    write the popped character to the screen
}

```

What is written to the screen for the input

i. ccaarrppeettss

ii. tattarrattat

a. Write a pseudo code for a program that takes **n** numbers as input and then arranges them in descending order of magnitude.

c. (i) Insert the elements 1, 0, 7, 3, 6, 8, 5, 9, 4, 2 into an initially empty binary search tree. The elements must be inserted in the given order.

(ii) Draw a picture of a binary search tree that will result if the value 8 is deleted from the given tree.

B4. [25 marks]

Suppose we want to handle the more general problem of counting the occurrences of all the words in some input. One solution is to keep the set of words seen so far sorted at all times, by placing each word into its proper position in the order as it arrives using a binary tree.

a. The following gives the content of the header file “tree.h” that contains the declaration of a node(it consist of a word and its count) and The binary tree itself.

```
#ifndef __TREE_H__
#define __TREE_H__
#include<string>
using namespace std ;
class Node
{
    public:
    string word ;
    int count ;
    Node * left ;
    Node * right ;
    Node(string s )
    {
        word = s ;
        count = 1 ;
    }
};

class Btree
{
    private :
    Node * root ;    // the root node

    public :
    Btree() ;
    void add(string item) ;    // add a given string to the binary tree
    void inorderPrint() ;    // produce an inorder print of items in binary tree
                                // for each node print the word and its counts
};
```

Provide the implementation of the public member functions.

NB : You can modify the class declaration to include other helper member functions.

a. With the aid of the Btree class , Implement a **main** function that's takes any arbitrary number of words as it input , count occurrence of each word and then produce an inorder traversal of the items(for each node print the word and its count) in the binary tree.

a. Write a short note on the following

I. Depth first traversal

II. Breadth first traversal

B5. [25 marks]

a. Given two arrays, **A** and **B**, each storing **m** and **n** integers, respectively, write the pseudocode for an algorithm **setDifference(A , B)** that returns a third array **C** that contains the integers in **A** that are not in **B**.

For example, if **A** = [7 , 4 , 9 , 12 , 2] and **B** = [5 , 12 , 3 , 7], then **C** should be [4 , 9 , 2].

a. Consider the following declaration of stack class that keeps track of an array of doubles.

```
class stack
{
    private :
        double * elements ; // this is a pointer to our elements
        int top ; // current position in the stack
        int size ; // size of stack
    public :
        stack(int s) ; // constructor
        void push(double a) ;
        double pop() ;
        bool isEmpty();
        bool isFull() ;
        double peep() ;
};
```

Provide Implementation of the member functions.

