FINAL EXAM
SECOND SEMESTER SESSION 2011/2012

COURSE CODE / NAME : STIA2023/DATA STRUCTURES AND ALGORITHM ANALYSIS
DATE : 13 JUNE 2012 (WEDNESDAY)
TIME : 2.30 – 5.00 P.M. (2 ½ HOURS)
VENUE : DTSO

INSTRUCTION :

1. This book script contains SECTION A for TEN (10) questions, SECTION B for ELEVEN (11) questions, and SECTION C for THREE (3) questions in FIFTEEN (15) printed pages excluding the cover page.
2. Answer ALL questions in the space provided.
3. You are NOT ALLOWED to remove the exam paper from the examination hall.

MATRIC NO : ____________________________________________
( with word )

IDENTIFICATION CARD NO. :

LECTURER : ____________________________________________

GROUP : ☐ TABLE NO. : __________

DO NOT OPEN THIS EXAMINATION PAPER UNTIL INSTRUCTED

CONFIDENTIAL
SECTION A: MULTIPLE CHOICE QUESTIONS (10 MARKS)
INSTRUCTION: Circle the best answer

1. A ____________ is a structure whose insert and erase operations occur at one end of a sequence called the top.
   a. Stack
   b. Queue
   c. List
   d. Linked list

2. By application of depth-first search strategy, an algorithm can perform a topological sort of a ____________ graph.
   a. directed acyclic
   b. directed cyclic
   c. undirected acyclic
   d. undirected cyclic

3. Insertions and deletions inside a contiguous list have ____________ running time.
   a. O(1)
   b. O(n)
   c. O(n^2)
   d. O(n^3)

4. A method, called something, has the following body
   
   ```
   switch(n)
   
   case 1:
     return 0;
   default:
     return 1 + something(n/3);
   
   ```

   What would be a correct signature for this method body? (Note: Correct means that it does not trigger any error or warning message at the compilation).
   a. public int something(int[] n)
   b. public double something(int[] n)
   c. public int something(int n[])
   d. public int something(int n)
5. A _____________ tree is an ordered tree that is full, except possibly for some missing nodes at the right of the bottom level.
   a. complete
   b. full
   c. empty
   d. binary

6. A full binary tree of height 9 has how many leaves?
   a. 512
   b. 511
   c. 81
   d. 80

7. What additional requirement is placed on an array, so that binary search may be used to locate an entry?
   a. The array elements must be integer numbers.
   b. The array must have at least 2 entries.
   c. The array must be sorted.
   d. The array's size must be a power of two.

8. When is Insertion sort a good choice for sorting an array?
   a. Each component of the array requires a large amount of memory.
   b. Each component of the array requires a small amount of memory.
   c. The array has only a few items out of place.
   d. The processor speed is fast.

9. Suppose we are sorting an array of eight integers using Quicksort, and we have just finished the first partitioning with the array looking like this:
   
   \[2 \ 5 \ 1 \ 7 \ 9 \ 12 \ 11 \ 10\]

   Which statement is correct?
   a. The pivot could be either the 7 or the 9.
   b. The pivot could be the 7, but it is not the 9.
   c. The pivot is not the 7, but it could be the 9.
   d. Neither the 7 nor the 9 is the pivot.
STIA2023 Data Structures and Algorithm Analysis

10. What is the best definition of a collision in a hash table?
   a. Two entries are identical except for their keys.
   b. Two entries with different data have the exact same key.
   c. Two entries with different keys have the same exact hash function.
   d. Two entries have different hash values.
SECTION B: STRUCTURED QUESTIONS (70 MARKS)
INSTRUCTION: Answer ALL questions at the space provided.

1. Explain ONE (1) advantage and ONE (1) disadvantage of the linked implementation of a queue relative to the contiguous implementation.

   (5 marks)

2. Given a directed graph as below:

   ![Graph Diagram]

   a) Show the steps to list the nodes of the graph in ascending order using breadth-first topological ordering.

   (3 marks)
b) List all paths from vertex A to vertex B and find the shortest path. (2 marks)

3. Based on the following algorithm:

```java
int sum = 0;
for (int k = 0; k <= n; k++)
    sum = sum - k / 2;
System.out.println("Sum: "+sum);
sum=0;
for (int k = 0; k <= n; k++)
    for (int j = 0; j <= n; j++)
        sum = sum - k / 2;
System.out.println("Sum: "+sum);
```

a) Show the steps to find the total operations of that algorithm. (4 marks)

b) What is the value of Big Oh? (1 mark)
Given the binary search tree (BST) below:

Redraw the BST after the following operations:

a) Delete node 17. (2 marks)

b) Add node 19 and node 27 (from the original tree). (3 marks)
5. Show the steps of inserting nodes 3, 2, 5, 10, 4, 15, 1
into initially empty AVL tree.
6. Why is binary search less efficient when written for Linked List than a normal sequential (linear) search?

(2 marks)

7. A newspaper route has recently been computerized. Information about each of the 100 customers is stored in individual records containing first name, last name, and payment due. In writing a computer program to process the customer records, the programmer is uncertain whether to add a procedure to sort the records.

a) If the records are NOT sorted, what is the maximum possible number of comparisons that must be made to obtain a particular customer's record using a sequential search?

(2 marks)

b) If the records are first sorted, what will be the maximum number of comparisons needed with a binary search to find a particular customer's record?

(2 marks)
8. Given a set of strings:

    win  had  her  and  you  his  them  get

Trace this set until all strings are available in ascending order by using Insertion sort algorithm.

(8 marks)

9. Given an array of integer values as follow:

    6, 9, 1, 10, 34, 12, 15, 8

a) Sort it using Merge Sort. Write down each step.

(7 marks)
b) What is the complexity of the Merge Sort algorithm? (1 mark)

10. Show the steps of HEAP SORT for the following array of elements in ascending order.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>60</td>
<td>70</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>50</td>
<td>10</td>
</tr>
</tbody>
</table>

(12 marks)
11. Draw a hash table with chaining and a size of 9. Use the hash function \( k \mod 9 \) to insert the keys 5, 29, 20, 0, and 18 into the table.

(8 marks)
SECTION C: PROGRAMMING QUESTIONS (20 MARKS)
INSTRUCTION: Answer ALL questions at the space provided.

1. The Java program below is about the implementation of a linked list. This implementation will use a Node class whose definition is given as follows:

```java
private class Node {
    Node next;
    Object data;

    public Node(Object _data) {
        next = null;
        data = _data;
    }
    public Node(Object _data, Node _next) {
        next = _next;
        data = _data;
    }
    public Object getData() {
        return data;
    }
    public void setData(Object _data) {
        data = _data;
    }
    public Node getNext() {
        return next;
    }
    public void setNext(Node _next) {
        next = _next;
    }
} // end class
```
You are required to complete the Java code below.

```java
public class LinkedList
{
    private Node head;
    private int listCount;

    // LinkedList constructor
    public LinkedList()
    {
        // this is an empty list, so the reference to the head Node is
        // set to a new node with no data
        listCount = 0;
    }

    public void add(Object data)
    // post: appends the specified element to the end of this list.
    {
        Node temp = new Node(data);
        Node current = head;
        // starting at the head node, crawl to the end of the list
        while(______________________________)
        {
            __________________________
        }
        // the last node's "next" reference set to our new node
        // increment the number of elements variable
        __________________________
    }

    public void add(Object data, int index)
    {
        // post: inserts the specified element at the specified
        // position in this list.
        Node temp = new Node(data);
        Node current = head;
        // crawl to the requested index or the last element in the
        // list, whichever comes first
        for(int i = 1; i < index && current.getNext() != null; i++)
        {
            ______________________________
        }
    }
```
// set the new node's next-node reference to this node's
// next-node reference
    temp.setNext(current.getNext());
// now set this node's next-node reference to the new node

// increment the number of elements variable

} 

} 

2. Consider the problem of printing the items in a binary tree. If the tree is empty, there is nothing to do. If the tree is non-empty, then it consists of a root and two subtrees. Print the item in the root and use recursion to print the items in the subtrees. Complete a method below that prints all the items on one line of output.

(6 marks)

static void __________________ ( TreeNode root ) {

    /* Print all the items in the tree to which root points.
       The item in the root is printed first, followed by the items
       in the left subtree and then the items in the right subtree*/

    if ( __________________ ) // Otherwise, there's nothing to
                                // print.
        {
            ____________________________ // Print the root item.

            preorderPrint( ____________ ); // Print items in left subtree.

            preorderPrint( ____________ ); // Print items in right subtree.

        }

} // end preorderPrint()
3. Write a recursive method, `numNodes(Node first)` to count the number of nodes, which takes a Node as a parameter and returns an integer. To begin, `current` is given the first node of the list. Regardless, `numNodes` checks to see if the list is empty. If so, it returns 0. Otherwise, it returns 1 plus the result of `numNodes` with input `current.getNext()`.

(6 marks)