CAMPBELL UNIVERSITY

NORTH CAROLINA, U. S. A.

ACADEMIC YEAR 2019/2020

JANUARY/FEBRUARY EXAMINATION

COMPUTER SCIENCE BACS2063(B)
DATA STRUCTURES AND ALGORITHMS

THURSDAY, 6 FEBRUARY 2020 TIME: 2.00 PM – 3.00 PM (1 HOUR)

BACHELOR OF SCIENCE DEGREE

Instructions to Candidates:

Answer **ALL** questions. All questions carry equal marks.

The code snippet in Figure 1 below shows the implementation of the Queue abstract data type (ADT) using linked implementation.

```
public class LinkedQueue<T> implements Queue<T>
{
      private Node firstNode;

      public LinkedQueue() {
      }

      public void add(T item){
       ...
      }

      public T remove(){
       ...
      }

      public boolean search(T item) {
            return search(firstNode, item)
      }

      private boolean search(Node currentNode, T item) {
            boolean found;
            if (currentNode == null)
                  found = false;
            else if (item.equals(currentNode.data))
                  found = true;
            else
                  found = search(currentNode.next, item);
            return found;
      }

      private class Node {
            private T    data;
            private Node next;

            private Node(T data) {
                  this.data = data;            }
      }
}
```

Figure 1: A recursive method for search with a private class Node

Figure 2 shows an example of a chain of linked nodes that stores 3 names.

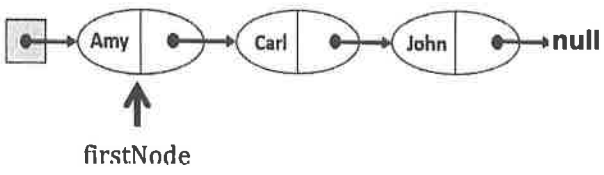

Figure 2: Linked nodes of names

BACS2063(B) DATA STRUCTURES AND ALGORITHMS

Question 1

a) Compare and contrast between the implementations of data structures using linked nodes and array. (4 marks)

b) In an abstract data type (ADT) specification, each operation shall include a
- brief description of what the operation does
- pre-condition
- post-condition
- returned data by the operation

Provide the specification for the **`add()`** method as shown in Figure 1. (4 marks)

c) Using only the `firstNode` reference, implement the **`add()`** method using Java code according to the ADT specification given in **Question 1 b)**. (8 marks)

d) Evaluate the time efficiency of the implementation of the **`add()`** method in **Question 1 c)** using the Big-O notation. Explain. (2 marks)

e) Suggest a better way to implement the **`add()`** method that can improve the time efficiency. (2 marks)

f) Based on Figure 1 and Figure 2, write a Java **`main()` method** to perform the following:
- Declare and create a queue object using the linked implementation in Figure 1.
- Add the 3 names as shown in Figure 2 into the queue.
- Use a loop to remove each name from the queue. Use `System.out.println()` to display the names.
- Write the output based on your program written above. (5 marks)

[Total: 25 marks]

Question 2

a) Perform a box trace to search for the target "**Lee**" based on Figure 1 and Figure 2. The traversal shall start with the first node, which is "Amy". For each **`search()`** method call, indicate the argument values, the statement(s) executed, and the returned value(s). (5 marks)

b) Use Figure 2 as an example, explain how **binary search** is done to search for the target "**Lee**". Clearly demonstrate all the steps. (5 marks)

Question 2 (Continued)

c) Figure 3 shows a list of integers:

7	5	3	8	2	1

Figure 3: A list of integers

Demonstrate how **Bubble Sort** algorithm is applied to produce the sorted list in ascending order. You are required to show the content of the list after each pass. (5 marks)

d) Construct a binary search tree according to the data given in Figure 3. Assume that the data are fed from index 0 to 5, i.e. left to right. (3 marks)

e) A hash function is given as h(key) = key % 7, where the size of the hash table is 7. Assume that separate chaining is used to resolve collisions. Draw the resulting hash table using the keys as shown in Figure 3. (7 marks)

[Total: 25 marks]