University of Swaziland

Department of Computer Science

Final Main Examination

MAY 2018

Title of paper : Data Structures and Algorithms

Course number : CSC311/CS342

Time Allowed : Three(3) hours

Instructions :

- *Each question carries 25 marks*
- *Answer any four (4) questions from questions 1 to 6.*

This paper may not be opened until permission has been granted by the invigilator

Question 1

(a) With the aid of a diagram, state precisely/formally the meaning of the statement: f(n) is O(g(n)) [3]

(b) With the aid of a well-labeled graph, compare the typical classes of running time functions. In each class give specific examples of algorithms that were discussed in this course. [8]

(c) Justify the following statement without directly invoking the definition of big-oh notation: N (N+ 1) is O (N^2). [2]

(d) With the aid of an example, distinguish between row major order and column major order allocation for a 2D array. [5]

(e) Write a general array mapping function, *Address (A[i,j])*, assuming column major order. Explain or show how you obtained this expression. [5]

(f) What is the big-oh time complexity for accessing element A[i,j]? Explain your answer. [2]

Question 2

(a) List and describe the operations of the stack data structure. [5]

(b) Compare an array-based implementation and linked-linked implementation of a stack. Which would you recommend as a better implementation? Justify your answer. [5]

(c) Using Java notation, write a **Stack** class that uses an array to implement a Stack. Provide code for all your member functions. [10]

(d) Using your implementation in (c) above, write a function that evaluates post-fix expressions of the form : 4 5 3 * + . Trace the execution of your function using this example. [5]

Question 3

(a) List and describe the operations of the queue data structure. [5]

(b) Using Java notation, write a generic **Queue** class that uses a circular linked-list to implement a queue structure. Provide code for all your member functions. [12]

(c) Analyze the Big-Oh time-complexity for each queue operation implemented in (b) above. [3]

(d) Based on your implementation in (c) above, write a function that finds and returns the largest value in a given queue of integer values. [5]

Question 4

Assuming $A\ [lo_1..hi_1,\ lo_2..hi_2]$ is an 2D array of Employee records and Employee class is as defined below:

```
class Employee
{  string  pin[6];
   string firstname[10];
   string lastname[20];
   int NumberofChildren;
 };
```

Using Java notation,

a) Define getters and setters for class Employee. What is the purpose of defining getters and setters? [7]

b) Write a function that, given the pin, firstname, lastname and number of children, inserts a new employee record into array A at a specific row index i and column index j. What is the running time complexity of this function? [5]

c) Write a function to sort the array A records by the personal identity number (pin). What is the running time complexity of this function? [10]

d) Override the **toString()** member function which returns a string representation of an employee record (first name, last name, pin number and number of children) . What is the running time complexity of this function? [4]

Question 5

(a) Using Java notation, define the structure of a binary search tree (BST). Your definition must contain two class definitions; class TreeNode - that models the structure of a node and class BSTree - that models the binary search tree. Show all data members, prototypes (function headers) of the required constructors and prototypes of member functions, but not the actual code of the constructors or member functions. [6]

(b) Using Java notation, write the constructor functions for classes TreeNode and BSTree defined in (a) above. [4]

(c) Using Java, and assuming your definition in (a) above, write code for the member functions:

(i) Adding a new value to a BST. [2]

(ii) Recursive Pre-order traversal of a BST. [2]

(d) Draw a binary search tree consisting of 13 nodes, and a height of 3. [4]

(e) Trace the execution of iterative post-order traversal algorithm on tree obtained in (d) above. [7]

Question 6 – 20 Marks

(a) Draw a sample weighted undirected graph containing 8 nodes and 15 edges. [4]
[*Use letters (A, B, C, etc) to label the nodes, and integer weights between 1 and 10 for the edges*]

(b) Distinguish between adjacency list and adjacency matric representation of a graph. Which representation would you recommend and why? [4]

(c) With the aid of the example given in (a) above, distinguish between the breath-first search (BFS) and depth first search (DFS) algorithms. [5]

(d) With the aid of the example given in (a) above, discuss Prim's minimum spanning tree algorithms. How does it compare with other (at least 1 other) minimum spanning tree algorithms. [6]

(e) With the aid of an example, discuss Dijkstra's shortest path algorithm. How does it compare with other (at least 2 other) shortest path algorithms [6]