Curtin University

Department of Computing

EXAMINATION

End of Semester 1, 2017

COMP1002 Data Structures and Algorithms

This paper is for Bentley Campus and Miri Sarawak Campus students

This is a CLOSED BOOK examination

Examination paper IS to be released to student

Examination Duration 2 hours

Reading Time 10 minutes

Notes may be written on the exam paper by Students during reading time

Total Marks 120

Supplied by the University

1 x 16 page answer book

Supplied by the Student

Materials

none

Calculator

No calculators are permitted in this exam

Instructions to Students

Student to attempt all questions. Answers to be written in blue or black pen in the supplied exam booklet.

For Examiner Use Only

Q	Mark
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	

Total _____

QUESTION ONE (Total: 10 marks): General

a) **(2 marks)** Why is Java less likely to run out of stack space than C/C++ in a recursive algorithm?

b) **(4 marks)**. Write a recursive function to calculate the factorial of N

ie: N! = N * (N-1) * (N-2)*…* 1, 1! = 1

c) **(4 marks)**. Explain why an array is less than ideal for implementing a FIFO queue. As part of your explanation, mention a typical (non-ideal) solution to implementing queues with an array and why it is less than ideal.

QUESTION TWO (Total: 25 marks): Sorting

a) Given the following array of numbers, show how MergeSort would sort them:
25, 20, 40, 5, 45, 50, 15, 10

i) **(7 marks)**. Draw a diagram of the splits and merges to be processed at every level.

ii) **(4 marks)**. On the diagram, write down how many **comparison** operations (ie: x<y) are involved at each merging branch.

b) **(2 marks)**. Why is it desirable to have an **in-place** sort?

c) Regarding QuickSort:

i) **(2 marks)**. What is the purpose of the pivot element in QuickSort?

ii) **(4 marks)**. Name and **describe** two (2) pivot selection strategies. For each strategy, your description must include how the strategy works, one (1) advantage and one (1) disadvantage **in comparison to the *other* strategy**.

d) **(6 marks)**. Bubble sort, insertion sort and selection sort are all $O(N^2)$ sorting algorithms. For each algorithm, give one (1) advantage and one (1) disadvantage in comparison to the others.

QUESTION THREE (Total: 25 marks): Lists, Iterators, Generics

a) **(6 marks)** List three variants on the basic linked list (single-ended, singly-linked) and for each variant describe how it differs from the basic version.

b) **(4 marks)** Give two (2) advantages and two (2) disadvantages common to all linked list variants in comparison to arrays.

c) **(12 marks)** Write the "QueueIterator" Java iterator class that traverses a linked list queue from head to tail.

Note that to satisfy the Iterator interface, you will need to at least implement methods hasNext(), next() and remove().

Assume the queue is made up of the following ListNode objects.

```
public class ListNode
{
  public double value = 0;
  public ListNode next = null;
}
```

You need to write this class:

```
public class QueueIterator implements Iterator
{
```

d) **(3 marks)** Give three reasons why it is a good idea to use generics over containing Objects when writing a general-purpose container class such as a linked list.

QUESTION FOUR (Total: 18 marks): Trees

a) **(4 marks)**. Draw the binary search tree that would result from inserting the following numbers in the order that they are shown:

25, 30, 40, 5, 10, 35, 37, 50, 45, 7

b) **(4 marks)**. For the tree created in part (a), draw the binary tree that would result if 40 is deleted.

c) **(4 marks)**. Using a diagram, describe what a degenerate binary search tree is, and why it is a problem.

d) **(6 marks)**. Given the partial definition of the BinaryTree and associated TreeNode classes below, write the method **max()** to find the maximum value in the tree. (Hint: the maximum value is the right-most node).

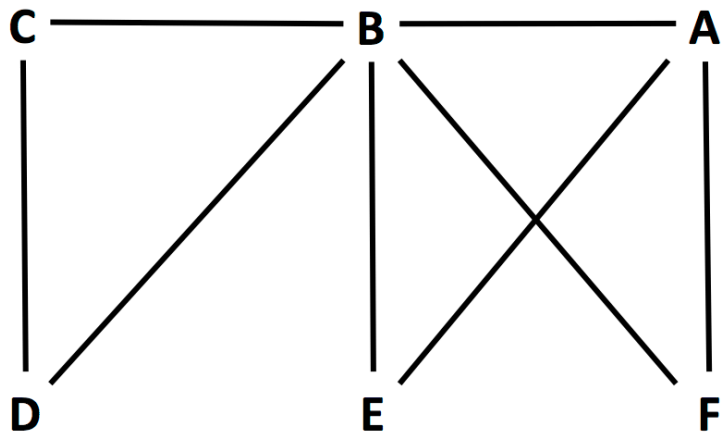
```
public class BinaryTree {
   // Inner class TreeNode
   private class TreeNode {
      public int value;
      public TreeNode left;
      public TreeNode right;
   }

   // Class BinaryTree
   private TreeNode root;

   public int max(){
      // You must implement this
   }

   // C'tors and other methods are not relevant to the question
}
```

QUESTION FOUR (Total: 12 marks): Graphs



a) **(3 marks)**. Provide the **adjacency matrix** representation for the above graph.

b) **(3 marks)**. Provide the **adjacency list** for each vertex of the above graph.

c) **(3 marks)**. Perform a **depth-first** traversal of the graph above, starting with vertex A. Select edges in alphabetical order. List the vertices in the order they are visited.

d) **(3 marks)**. Perform a **breadth-first** traversal of the graph above, starting with vertex A. Select edges in alphabetical order. List the vertices in the order they are visited.

QUESTION FIVE (Total: 20 marks): Heaps

a) Given the following list of numbers:

39, 11, 7, 23, 18, 42, 8, 3

i) **(7 marks)**. Draw the **heap** (as a tree diagram) that would be built if the above numbers were inserted into a **min heap** in the order they are listed.

ii) **(2 marks)**. Convert the heap tree in your answer from part i) into an **array** representation of the tree.

iii) **(3 marks)**. Show what the **array** form of the heap would look like after inserting the value of **6** into the heap (hint: use your tree to help you trace the 'trickle-up').

b) **(3 marks)**. Give three (3) differences between a heap and a binary search tree.

c) **(5 marks).** Is there any difference in the time complexities for max and min heap when inserting data that is in **descending** order? Justify and include Big-O notation time complexities in your answer.

QUESTION SIX (Total: 10 marks): Hashing

a) **(4 marks)**. Given the following hash function that maps an integer key to an index given a table of length 11:

```
public int hash(int key) {
   int hashIdx;
   hashIdx = key % 11;
}
```

Use the above hash function to insert the following key **:** value pairs into the hash table below (copy the table to your exam booklet). Handle collisions using the **quadratic probing** method.

18 : “First” 25 : “Second” 7 : “Third” 14 : “Fourth” 3 : “Fifth”

Index	*Key*	*Value*
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

b) **(3 marks)**. Explain why collisions are a problem in hash tables (use Big-O notation to illustrate your explanation).

c) **(3 marks).** Describe how the **double-hashing** collision handling algorithm works.

END OF EXAMINATION