Venue _____

Student Number |__|__|__|__|__|__|__|__|

Family Name _____

First Name _____

Curtin University

Department of Computing

EXAMINATION

End of Semester 2, 2016

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 in the margins of exam paper may be written by Students during reading time

Total Marks 100

Supplied by the University

1 x 8 page answer book

Supplied by the Student

none

No calculators are permitted in this exam

Instructions to Students

Students to answer all questions in the space provided within the exam paper.

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: 20 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)**. Consider the following recursive function to calculate the factorial of N (ie: N!)

```
public int factorial(int N) {
      if (N == 1) {
         return 1;
      }
      else {
         return N * factorial(N - 1);
      }
   }
```

Rewrite the function into an *iterative* form (ie: with looping rather than recursion)

c) **(3 marks)**. Explain why an array is less than ideal for implementing a FIFO queue. As part of your explanation, mention a typical solution to implementing queues with an array and why it is less than ideal.

d) **(4 marks)** Using a diagram, describe the difference between primary and secondary clustering in a hash table.

e) **(4 marks)** Consider that you are asked to write a search function to extract the birthdate of all people over the age of 65 this year (ie: born before 01-Jan-1946). The search results will be used to perform a one-off statistical analysis of people eligible for pension, hence the extraction only needs to occur once. The dataset is large (millions of records), exists in a file on disk and is not sorted. Would you use binary search or linear search to find the people? Justify your answer in terms of time complexity (include a Big-O analysis as part of your answer).

f) **(3 marks)**. Describe the purpose of Java's Object Serialization schema.

QUESTION TWO (Total: 25 marks): Sorting

a) **(3 marks)**. Given the following list of numbers, which $O(N^2)$ sorting algorithm will be faster to sort it: Bubble Sort, Selection Sort or Insertion Sort? Justify your answer.

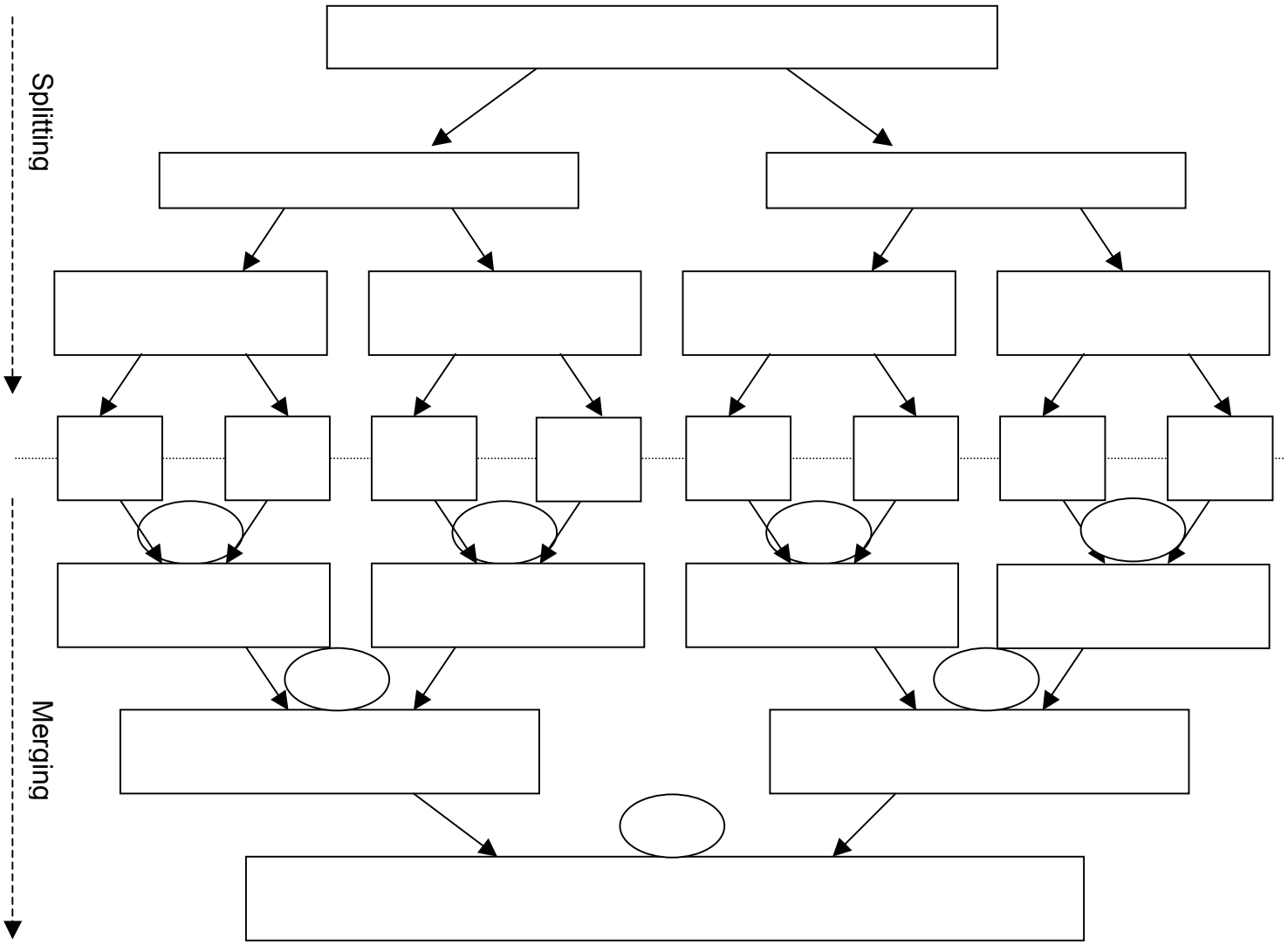
5, 10, 22, 80, 6, 87, 90

b) Given the following array of numbers, show how MergeSort would sort them:
22, 27, 4, 64, 32, 3, 53, 49

i) **(3 marks)**. Draw the recursion tree (below) in your answer book. Fill in the "Splitting" boxes depicting the splits and sub-arrays to be processed at each branch (do not show the merging of the sub-arrays yet though).

ii) **(3 marks)**. When the recursion unwinds, MergeSort will begin merging the different sub-arrays, comparing elements as it merges. Complete your diagram from part (i) so that merges are shown in the "Merging" boxes.

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



QUESTION TWO continued: Sorting

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) **(10 marks)** Write the "insertionSort" recursive Java method that implements Insertion Sort using a linked list and returns the (head of the) sorted list. The only assumptions you can make about the incoming list are that it will be non-null and that the elements will be real numbers (ie. not NaN, positive infinity or negative infinity.

Assume the list is made up of the following ListNode objects:

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

You need to write this method:

```
public ListNode insertionSort(double [] inArray)
{
```

b) **(12 marks)** Write the "TreeIterator" Java iterator class that traverses a binary tree in-order.

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

Assume the tree is made up of the following TreeNode objects.

```
public class TreeNode
{
  public int value = 0;
  public TreeNode leftChild = null;
  public TreeNode rightChild = null;
}
```

You need to write this class:

```
public class TreeIterator implements Iterator
{
```

c) **(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: 30 marks): Trees

a) **(3 marks)**. Draw the binary search tree that would result from inserting the following numbers in the order that they are shown:

36, 42, 21, 25, 52, 48, 15, 39, 50, 3

b) **(3 marks)**. For the above tree, draw the binary tree that would result if 42 was deleted.

c) **(4 marks)**. A B-Tree can have any number of data items per node. Discuss in detail how the number of data items is chosen for a B-Tree.

d) **(6 marks)**. Draw the 2-3-4 search tree that results in inserting the following sequence of numbers, {9, 52, 36, 39, 10, 90, 64, 13} into an initially empty tree (i.e. one tree per insertion, showing all of your working).

e) **(3 marks)**. Convert your answer from 4(d) above into a Red-Black tree.

f) **(2 marks)**. What is a degenerate binary search tree and why is it a problem?

g) **(3 marks)**. Inserting a new value into a binary search tree is at best O(log N) and at worst O(N), but this depends on how the tree is structured. What kind of tree will guarantee an O(log N) case for insert even in the worst case for that tree? Why?

h) **(2 marks)**. What is the successor node when deleting a value in a binary search tree and what role does it play?

i) **(4 marks)**. Given the partial definition of the BinaryTree and associated TreeNode classes below, write the method min() to find the minimum value in the tree. (Hint: the minimum value is the left-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 min(){
      // You must implement this
   }

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

QUESTION FIVE (Total: 20 marks): Heaps

a) Given the following list of numbers:

39, 11, 7, 23, 62, 42, 19, 3

i) **(7 marks)**. Draw the heap (as a tree diagram) that would be built if the above numbers were inserted into a max 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 40 into the heap (hint: use your tree to help you trace the 'trickle-up').

b) **(3 marks).** A heap is a type of binary tree, but one that can be implemented using either:

- An actual binary tree data structure OR
- An array organised such that it represents the tree

Explain why it is more common to implement heaps using the array form.

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.

END OF EXAMINATION