UNIVERSITY OF SOUTHAMPTON COMP1009W1

SEMESTER 2 EXAMINATION 2009/2010

DATA STRUCTURES AND ALGORITHMS

Duration: 120 mins

You must enter your Student ID and your ISS login ID (as a cross-check) on this page. You must not write your name anywhere on the paper.

Student ID:	
ISS ID:	

Question	Marks
1	
2	
3	
4	
Total	

Answer THREE *questions out of* FOUR.

This examination is worth 85%. The tutorials were worth 15%.

University approved calculators MAY be used.

Each answer must be completely contained within the box under the corresponding question. No credit will be given for answers presented elsewhere.

You are advised to write using a soft pencil so that you may readily correct mistakes with an eraser.

You may use a blue book for scratch—it will be discarded without being looked at.

Question 1

(a) Give five container classes that are part of the Java collection and describe briefly what they do and how they are implemented. *(10 marks)*

1

2

3

4

5

The algorithm for bubble sort (not covered in the course) to sort an array $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})$ is given by

```
BUBBLESORT(a)
  for i←0 to n-2
    for j←0 to n-2-i
      if a_{j+1} < a_j
         swap a_j  and a_{j+1}
       endif
     endfor
  endfor
```

(b) Write a Java method to performing bubble sort on an array of integers. *(7 marks)*

(c) Show how you could make your sort program generic so that it will sort an array of comparable objects. *(5 marks)*

(d) What is the time complexity of bubble sort? Explain your answer. *(3 marks)*

(e) Java uses quick sort for sorting arrays of primitive types and merge sort to sort array lists. Explain why Java makes this choice. In your answer you should make reference to the time complexity of the algorithms and whether the algorithms are stable and in-place. *(8 marks)*

End of question 1

Q1: (a) $\frac{\quad}{10}$ (b) $\frac{\quad}{7}$ (c) $\frac{\quad}{5}$ (d) $\frac{\quad}{3}$ (e) $\frac{\quad}{8}$ Total $\frac{\quad}{33}$

• Do not write in this space •

Question 2 Merge sort has the form

```
MergeSort(a[1:n]) {
  if (n>1) {
    b ← a[1:n/2]
    c ← a[n/2+1:n]
    MergeSort(b)
    MergeSort(c)
    Merge(b,c,a)
    }
  }
```

The number of comparison operations to merge two arrays of length $n/2$ is n.

(a) Let $T(n)$ be the number of comparison operations. Write down a recurrence relation for $T(n)$ valid if $n = 2^m$ *(4 marks)*

$T(n) =$

(b) Write down the boundary condition $T(1)$ and use the recurrence relation to compute $T(2)$, $T(4)$, and $T(8)$ *(4 marks)*

$T(1) =$

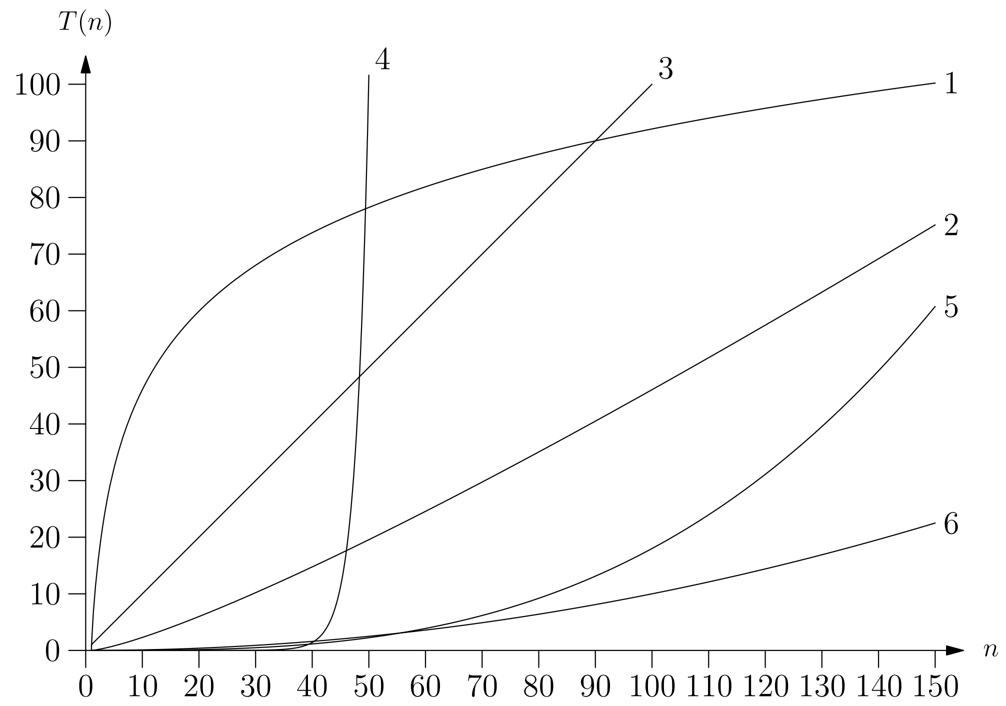
$T(2) =$

$T(4) =$

$T(8) =$

(c) Demonstrate, for $n = 2^m$, that $f(n) = n \log_2(n)$ satisfies the recurrence relation in part (a) *(6 marks)*

(d) The graph below shows the time complexity for the following algorithms (a) $\Theta((n/a)!)$, (b) $\Theta(n^2)$, (c) $\Theta(n\,\log(n))$, (d) $\Theta(n)$, (e) $\Theta(n^3)$, and (f) $\Theta(\log(n))$. Match the time complexity classes with the curves on the graph.



(6 marks)

1.	**2.**
3.	**4.**
5.	**6.**

• Do not write in this space •

(e) Which of the following statements are true? Give reasons why (marks will only be awarded if correct reasons are given). *(8 marks)*

(i) All $\Theta(n^2)$ algorithms are faster than all $\Theta(n^3)$ algorithms

(ii) An $O(n)$ algorithm will run faster than a $\Omega(n\,\log(n))$ algorithm for sufficiently large n

(iii) All $O(n^3)$ algorithms run slower than all $O(n^2)$ asymptotically

(iv) A $\Theta(n!)$ algorithm runs slower than any exponential algorithm in the limit of large n

(f) Why is it widely believed that $NP \neq P$? *(5 marks)*

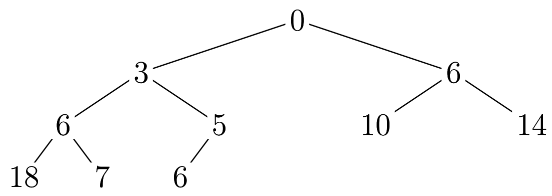
End of question 2

Q2: (a) $\frac{}{4}$ (b) $\frac{}{4}$ (c) $\frac{}{6}$ (d) $\frac{}{6}$ (e) $\frac{}{8}$ (f) $\frac{}{5}$ Total $\frac{}{33}$

• Do not write in this space •

TURN OVER

Question 3 Consider the **heap** represented as a binary tree

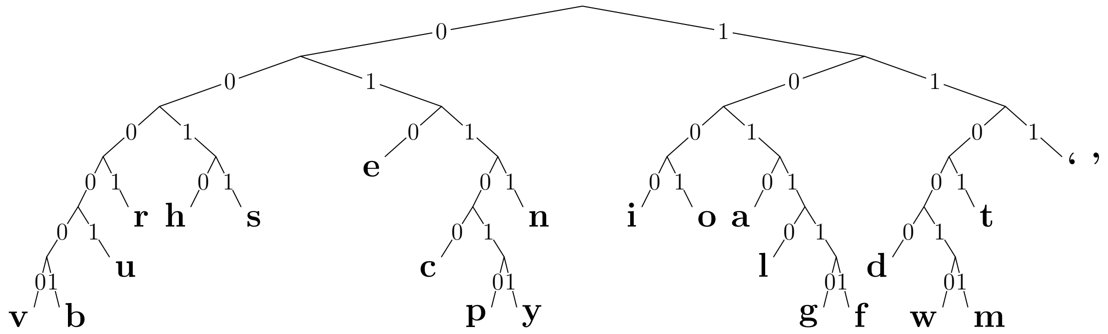


(a) Show how the heap would be stored in the computer memory *(3 marks)*

(b) Draw a binary tree representing the heap after we add 4 to it *(5 marks)*

(c) Draw a binary tree representing the heap above when we run `removeMin()` *(5 marks)*

(d) Given the Huffman tree



decode 001110110000011100010111001 0 *(6 marks)*

(e) Give a high level description (ignoring implementation details) of how a Huffman tree is constructed for the set of English letters. *(8 marks)*

(f) Describe how a heap is used to construct the Huffman tree *(4 marks)*

(g) Describe how a heap is used in Heap Sort *(2 marks)*

End of question 3

Q3:	(a) $\frac{\quad}{3}$	(b) $\frac{\quad}{5}$	(c) $\frac{\quad}{5}$	(d) $\frac{\quad}{6}$	(e) $\frac{\quad}{8}$	(f) $\frac{\quad}{4}$	(g) $\frac{\quad}{2}$	Total $\frac{\quad}{33}$

• Do not write in this space •

Question 4

(a) Draw the binary search tree which results from inserting the following list of numbers into the tree

$$84, 39, 78, 79, 91, 19, 33, 76, 27, 55.$$

(5 marks)

(b) Draw the tree after you delete 27. *(3 marks)*

(c) Draw the tree after you delete 76. *(3 marks)*

(d) Draw the tree after you delete 39. *(4 marks)*

(e) Derive the typical and worst case time complexities for insertion. Briefly explain the strategies used to reduce the worst case time complexity. *(10 marks)*

(f) Explain why binary search trees are commonly used to represent sets. What alternative data structure is also used for this purpose? How do these data structures compare? *(8 marks)*

End of question 4

Q4: (a) $\frac{\quad}{5}$ (b) $\frac{\quad}{3}$ (c) $\frac{\quad}{3}$ (d) $\frac{\quad}{4}$ (e) $\frac{\quad}{10}$ (f) $\frac{\quad}{8}$ Total $\frac{\quad}{33}$

END OF PAPER