UNIVERSITY OF SOUTHAMPTON COMP1009W1

SEMESTER 2 EXAMINATION 2008/2009

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 page.

| | |
|---|---|
| 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.*

Number of pages 17

**Question 1** Below we show part of a binary search tree class for storing integers

```
public class BinarySearchTree {
   Node root;
   int number_of_elements;

   private static class Node {
      int element;
      Node left;
      Node right;
      Node parent;
   }

}
```

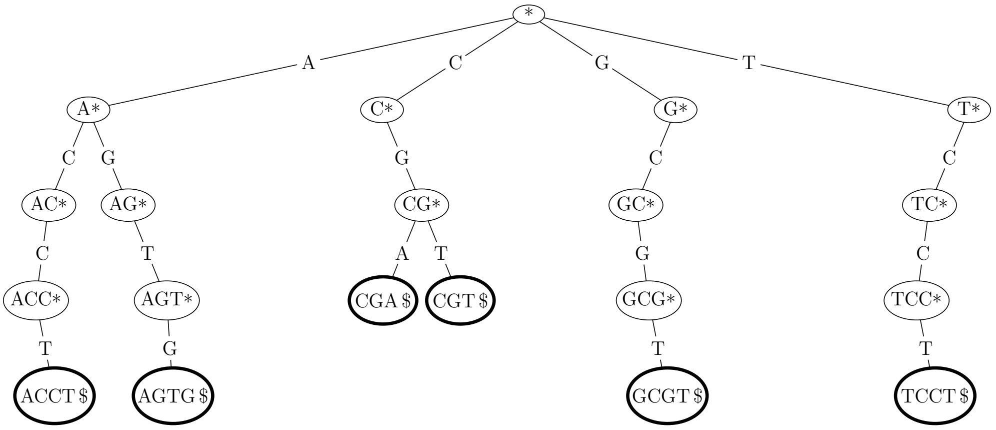
(a) Write a recursive method `sum()` which sums all the elements stored in the tree. Approximately correct code is sufficient. *(5 marks)*

(b) Show how you would make the BinarySearchTree into a generic class. *(5 marks)*

• Do not write in this space •

(c) Write a method `getNode(E obj)` which returns the node with element `obj` if it exists otherwise it returns `null`. You may assume `E` extends the Comparable interface. *(8 marks)*

(d) A digital tree or trie is a multi-way tree which can also be used to implement a set. An illustration of a trie that could be used for storing DNA sequences is shown below. This is a 4-way trie storing the sequences “ACCT$”, “AGTG$”, “CGA$”, “CGT$”, “GCGT$” and “TCCT$”. Describe how the tree can be modified to be more efficient (turned into a Patricia trie).



*(4 marks)*

• Do not write in this space •

(e) Give the retrieval time of an element from a `TreeSet`, a `HashSet` and a Trie. For full marks give a brief explanations of your answer. *(6 marks)*

1

2

3

• Do not write in this space •

Pseudo code for a possible hashing function for DNA sequences is shown below

```
hash = 17;
foreach base b in a sequence
  hash = 31*hash + integer(b)
return hash
```

where *integer(b)* turns the bases, A, C, G, T, into integers 0, 1, 2, 3.

(f) Explain, why a `HashSet` may not be any faster than the `Trie` at inserting or retrieving data. *(5 marks)*

End of question 1

Q1: (a) $\frac{\quad}{5}$ (b) $\frac{\quad}{5}$ (c) $\frac{\quad}{8}$ (d) $\frac{\quad}{4}$ (e) $\frac{\quad}{6}$ (f) $\frac{\quad}{5}$ Total $\frac{\quad}{33}$

• Do not write in this space •

**Question 2** The Towers of Hanoi problem is solved by the program

```
hanoi(n, A, B, C)
{
   if (n>0) {
     hanoi(n-1, A, C, B);
     move(A, C);
     hanoi(n-1, B, A, C);
   }
}
```

(a) Let $T(n)$ be the number of times `move` is called to solve the Hanoi problem of size $n$. Write down a recurrence relation for $T(n)$. *(4 marks)*

$T(n) =$

(b) Write down the boundary condition $T(1)$ and use the recurrence relation to compute $T(2)$, $T(3)$, and $T(4)$ *(4 marks)*

$T(1) =$

$T(2) =$

$T(3) =$

$T(4) =$

(c) Prove by induction that, $f(n) = 2^n - 1$ satisfies the recurrence relation in part (a) *(8 marks)*

Here is an algorithm for performing sort

```
public static void Sort(double[] data) {
    for (int i = 1; i < data.length; i++) {
        if (data[i]<data[i-1]) {
            double temp = data[i];
            int j = i;
            do {
                data[j] = data[j-1];
                j--;
            } while (j>0 && data[j-1] > temp);
            data[j] = temp;
        }
    }
}
```

(d) Give upper and lower bounds on the time complexity of the algorithm? Explain your reasoning. *(9 marks)*

(e) Briefly explain how binary search works and explain its time complexity *(8 marks)*

1

2

End of question 2

Q2: (a) $\frac{\quad}{4}$ (b) $\frac{\quad}{4}$ (c) $\frac{\quad}{8}$ (d) $\frac{\quad}{9}$ (e) $\frac{\quad}{8}$ Total $\frac{\quad}{33}$

• Do not write in this space •

**Question 3**

(a) Briefly describe the quick sort algorithm. *(5 marks)*

(b) Explain why choosing a pivot for quick sort is crucial to the performance of the algorithm and explain a common method for choosing a good pivot. *(6 marks)*

(c) What is the worst case and average case time complexity of quick sort. Give a justification for your answer. *(6 marks)*

• Do not write in this space •

(d) Explain why no algorithm using binary comparisons can sort an array in less than $\Theta(n \log(n))$ operation on average. *(10 marks)*

• Do not write in this space •

(e) Describe very briefly how radix sort works on a set of integers. *(4 marks)*

(f) What is the time complexity of radix sort? Why does this not contradict part (c) *(2 marks)*

1

2

End of question 3

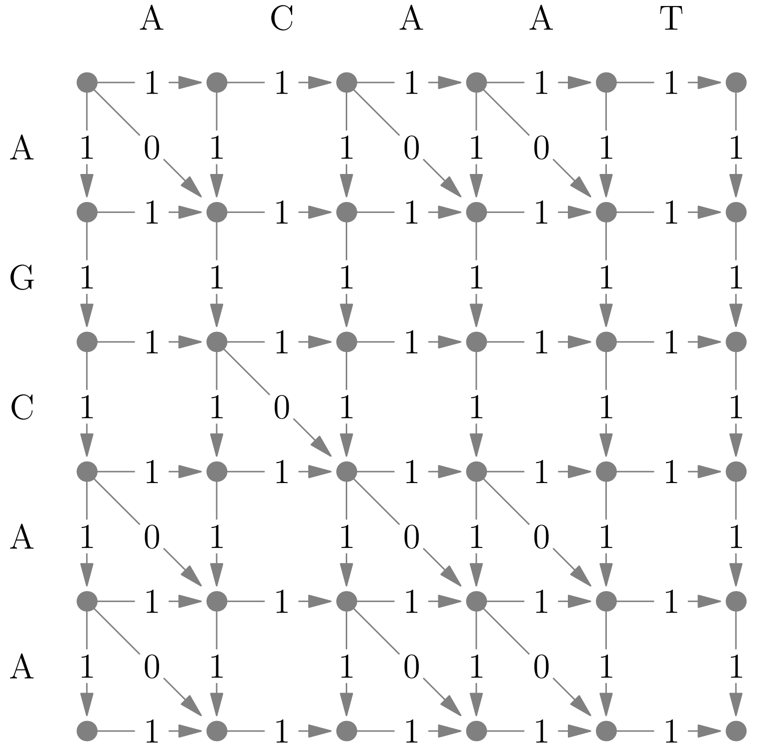
Q3: (a) $\frac{\quad}{5}$ (b) $\frac{\quad}{6}$ (c) $\frac{\quad}{6}$ (d) $\frac{\quad}{10}$ (e) $\frac{\quad}{4}$ (f) $\frac{\quad}{2}$ Total $\frac{\quad}{33}$

**Question 4**

(a) Explain the dynamic programming strategy. *(5 marks)*

• Do not write in this space •

(b) The edit distance between two strings is the number of edit operations needed to go from one string to another. In the simplest case we can consider the only edit operation to be insertion. In this case, the edit distance between the string ”AAC” and “ACT” is two as each string would have to have an insertion to make them identical (i.e. both strings could be turned into “AACT”). The edit distance can be computed by finding the lowest cost path on a edit graph starting from the top left hand corner and ending at the bottom right hand corner. The edit graph for two strings “ACAAT” and “AGCAA” is shown below. Using dynamic programming compute the costs of reaching each node (write the costs next to the nodes). Using the backwards algorithm to find the optimal path (show this on the graph). Write down the closest common string.



Closest Common String: ______________

*(16 marks)*

(c) Briefly describe the following strategies, giving examples of their use. 1) Brute force, 2) Branch and bound, 3) Divide and conquer and 4) Greedy method. *(12 marks)*

1

2

3

4

End of question 4

Q4: (a) $\frac{\quad}{5}$ (b) $\frac{\quad}{16}$ (c) $\frac{\quad}{12}$ Total $\frac{\quad}{33}$

**END OF PAPER**