UNIVERSITY OF SOUTHAMPTON COMP1009W1

SEMESTER 2 EXAMINATION 2006/2007

DATA STRUCTURES AND ALGORITHMS

Duration: 120 mins

Answer THREE *questions*

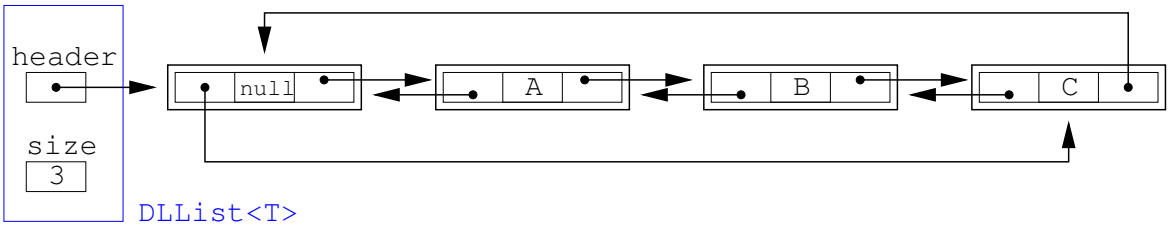
This examination is worth 85%. The tutorials were worth 15%.

University approved calculators MAY be used.

 Number of pages 6

Question 1

A doubly linked list consists of nodes that can point in either direction. An efficient implementation uses a dummy node which contains no data. An example of a doubly linked list is shown below.



(a) Write a generic class definition for the doubly linked list described above including the instance variables, a private nested class for the nodes and a default constructor. Do not include any other methods. Approximately correct code is acceptable. *(10 marks)*

(b) Write a `boolean add(T o)` method which adds element `o` to the end of the list (in the example above it would add the element after `C`). *(5 marks)*

(c) Describe the `Iterator<T>` interface of the Java Collect class. *(3 marks)*

(d) Write an `Iterator<T> iterator()` method which returns a private nested class of type *DLListIterator* *(2 marks)*

(e) Write a private nested class `DDListIterator()` which implements the methods of the `Iterator<T>` interface as well as a constructor method. *(13 marks)*

Question 2

(a) What is the time complexity of the following programs (written is pseudo code)

```
prog1(a,b,c)
{
  for (i=1; i<n; i++) {
    for (j=1; j<i; j++) {
       for (k=0; k<100; k++) {
          a(i,j) = b(j)*c(i,k)
       }
    }
  }
}

prog2(a,b,c)
{
  for (i=1; i<n; i++) {
    for (j=1; j<n; j++) {
      if (i-j=1 || i-j=-1)
       for (k=0; k<n; k++) {
          a(i,j) = b(j)*c(i,k)
       }
    }
  }
}
```

Explain your answers. *(6 marks)*

(b) If a program takes 1s on an input of size $n = 1000$, how long will take on an input of size $n = 10\,000$ if the time complexity is

(i) logarithmic, i.e. $\Theta(\log(n))$

(ii) log-linear, i.e. $\Theta(n \log(n))$

(iii) quadratic, i.e. $\Theta(n^2)$

TURN OVER

(iv) cubic, i.e. $\Theta(n^3)$

Show your working. *(8 marks)*

(c) Which of the following statements is true?

(i) A $\Theta(n\log(n))$ algorithm will always beat a $\Theta(n)$ algorithm

(ii) An $O(n^2)$ algorithm will always run faster than an $O(n^3)$ algorithm for sufficiently large n

(iii) An $O(n^2)$ algorithm will run faster than an $\Omega(n^2\log(n))$ algorithm for sufficiently large n

Explain your answer. *(6 marks)*

(d) The Fibonacci number, f_n, is generated from the recursion relation $f_n = f_{n-1} + f_{n-2}$ with $f_1 = f_2 = 1$ write an **efficient** program to calculate f_n. What is the time complexity of your code?

(4 marks)

(e) Consider the program below for computing Fibonacci numbers

```
Fibonacci(n)
{
  if (n<3) return 1;
  return Fibonacci(n-1) + Fibonacci(n-2)
}
```

Let $T(n)$ be the number of additions needed for the above program to compute f_n. Write a recursion relation for $T(n)$ and provide appropriate base cases. *(4 marks)*

(f) Show that $T(n) \in \Theta(\phi^n)$, by substitution, $c\phi^n$ into the recursion formula and ignoring non-leading terms (i.e. terms small compared to ϕ^n). Compute, ϕ. Asymptotically how much longer does it take to compute $T(n+10)$ compared to $T(n)$? *(5 marks)*

Question 3

(a) Show how Merge Sort would sort the numbers 97, 88, 43, 1, 43, 81, 21, 45. *(8 marks)*

(b) What does it mean for a sort algorithm to be stable? Why is it desirable for a sort algorithm to be stable? Is Merge Sort stable? *(6 marks)*

(c) Derive an expression for the time complexity of Merge Sort. *(8 marks)*

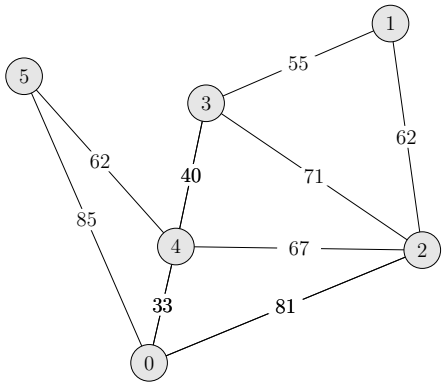
(d) Explain why the choice of the pivot is crucial to the success of Quick Sort. What problem occurs if we use the first element as the pivot? *(6 marks)*

(e) What is the worst case time complexity of Merge Sort and Quick Sort? Which sorting algorithm is preferred in practice and why? *(5 marks)*

Question 4

(a) Describe Prim's algorithm for computing the minimum spanning tree. Explain how this implemented efficiently. *(8 marks)*

(b) Compute the minimum spanning tree for the graph shown below.



Starting from node 0, show the order in which the minimum spanning tree is constructed using Prim's algorithm.

(8 marks)

(c) Describe a possible application of minimum spanning trees.

(2 marks)

(d) What do we mean by an "efficient" algorithm? Give two examples of efficient graph algorithms (not including minimum spanning tree). Describe two graph problems for which there are no known efficient algorithms. *(6 marks)*

(e) Describe in outline how branch and bound works. Is branch and bound an "efficient algorithm"? Under what conditions is branch and bound useful? *(9 marks)*

END OF PAPER