University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER I EXAMINATION – 2011/2012

COMP 20010/20210

Data Structures & Algorithms I

Prof. ?

Mr. J. Dunnion

Dr Eleni Mangina

Time allowed: 2 hours

Instructions for candidates

Answer **any** two questions. All questions carry equal marks (50).

Use of calculators is prohibited.

Instructions for invigilators

Use of calculators is prohibited.

1. Answer parts (a) to (f). **50 marks in total**

(a) (5 marks)

Briefly define the following variable modifiers in terms of scope (or visibility) of instance variables: public, protected, private, static, final.

(b) (10 marks)

Briefly define the following terms in the context of object-oriented design and provide an example: Inheritance and polymorphism.

(c) (5 marks)

Using examples describe the purpose of Java's interface and implements constructs.

(d) (5 marks)

You have been hired to build an object-oriented Java program to manage University's grade book. All **students** have a **name** and **type** (represented as Strings). For **undergraduate** students, the database records the number of years until graduation (an integer). For **postgraduate** students the database stores the years of study (an integer). Finally, a student's **description** method returns the concatenation of the student's name and type separated by space. Write Java code that implements these three classes.

(e) (10 marks)

In the following program, called ComputeResult, what is the value of result after each numbered line executes?

```
public class ComputeResult {
    public static void main(String[] args) {
            String original = "software";
            StringBuffer result = new StringBuffer("hi");
            int index = original.indexOf('a');
    /*1*/   result.setCharAt(0, original.charAt(0));
    /*2*/   result.setCharAt(1, original.charAt(original.length()-1));
    /*3*/   result.insert(1, original.charAt(4));
    /*4*/   result.append(original.substring(1,4));
    /*5*/   result.insert(3, (original.substring(index, index+2) + " "));
            System.out.println(result); }}
```

(f) (15 marks)

Consider the following implementation of a class Square

```
public class Square
{       private int sideLength;
        private int area;
        public Square(int initiallength)
        {
                sideLength = initiallength;
                area = sideLength * sideLength;
        }

        public int getArea ()  {  return area; }
        public void grow ()  {  sideLength = 2 * sideLength;}
}
```

What error does this class have? How would you fix it?

2. Answer parts (a) to (c). 50 marks in total

(a) (20 marks)

(i) A technique that can be used to describe the running time of an algorithm is known as *asymptotic notation*. One form of this technique is known as "Big-Oh" notation. Give the definition for this notation. **(4 marks)**

(ii) In the following, use either a direct proof (by giving values for c and n_0 in the definition of "Big-Oh" notation) or cite one of the rules given in the textbook or in the lecture slides:

Show that if f(n) is O(g(n)) and d(n) is O(h(n)), then f(n)+d(n) is O(g(n)+h(n)). **(6 marks)**

(iii) In the following, use either a direct proof (by giving values for c and n_0 in the definition of big-Oh notation) or cite one of the rules given in the textbook or in the lecture slides:

Show that $3(n+1)^7 + 2n\log n$ *is* $O(n^7)$. **(4 marks)**

(iv) Algorithm A executes *10nlogn* operations, while algorithm B executes n^2 operations. Determine the minimum integer value n_0 such that A executes fewer operations than B for $n \geq n_0$ **(6 marks)**

(b) (10 marks)

(i) When comparing algorithms, the typical approach adopted is to evaluate each algorithm's performance in terms of "Big-Oh" notation and then to compare performances based on a "hierarchy of functions". What do we mean by "hierarchy of functions" and how is it used to compare algorithms? **(4 marks)**

(ii) Based on the properties of the "Big-Oh" notation provide the running times of the following algorithms: **(6 marks)**

$5n^4 + 3n^3 + 2n^2 + 4n + 1$

$5n^2 + 3n\log n + 2n + 5$

$20n^3 + 10n\log n + 5$

$3\log n + 2$

2^{n+2}

$2n + 100\log n$

(c) (20 marks)

Another way of carrying out the analysis of an algorithm is to estimate the running time of the algorithm. To do this, we identify a set of *primitive operations* to which we assign a fixed and equivalent running time. List the seven primitive operations that we used in the course and work out the running time of the following pseudo code algorithms. Which algorithm is better? Explain your answer.

```
Algorithm PrefixAverages1(A, n):
    Input:      An integer array A of size n.
    Output:     An array X of size n such that X[j] is the
                average of A[0], ..., A[j].
    Let X be an integer array of size n
    for j=1 to n-1 do
        a ← 0
        for k=1 to j do
                a ← a + A[k]
        X[j] ← a / (j+1)
    return X
```

```
Algorithm PrefixAverages2(A, n):
    Input:      An integer array A of size n.
    Output:     An array X of size n such that X[j] is the
                average of A[0], ..., A[j].
    Let X be an integer array of size n
    runningSum ← 0
    for j=0 to n-1 do
        runningSum ← runningSum + A[k]
        X[j] ← runningSum / (j+1)
    return X
```

3. Answer parts (a) to (e). **50 marks in total**

(a) **(10 marks)**
What is a Vector? List the operations that are associated with this ADT.

(b) **(10 marks)**
A key feature of Vectors is their use of extendable arrays. Explain the concept of an extendable array and list the steps that must be performed to extend an array when inserting a new element into a Vector.

(c) **(10 marks).**
Give the algorithm for insertion at a given rank into a Vector that uses extendable arrays.

(d) **(10 marks).**
What is a List, and how does it differ from a Vector? Describe briefly how the List ADT can be implemented using a doubly linked list. Illustrate your answer by drawing boxes and lines diagrams that show the following: a list containing the strings “London”, “New York”, and “Paris”; and an empty list.

(e) **(10 marks).**
Compare the array-based and linked-list based implementation strategies of a Vector in terms of their runtime performance. Which one is better and why?

4. Answer parts (a) to (c). 50 marks in total

(a) (20 marks)

(i) What is a *Stack*? List the operations commonly associated with the Stack Abstract Data Type (ADT). (**5 marks)**

(ii) Describe the output of the following series of stack operations on a single, initially empty stack: *push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop().* **(4 marks)**

(iii) Describe in pseudo code a linear-time algorithm for reversing a queue Q. To access the queue, you are only allowed to use the methods of queue ADT. *Hint: Consider using a Stack.* **(5 marks)**

(iv) Describe how to implement two stacks using one array. The total number of elements in both stacks is limited by the array length; all stack operations should run in O(1) time. **(6 marks)**

(b) (15 marks)

(i) What is a *Queue*? List, giving a brief description for each one, the operations commonly associated with the Queue Abstract Data Type (ADT). **(5 marks)**

(ii) What is a *Deque*? Give the pseudo-code algorithm for insertion into the front of a Deque. Show, using diagrams, how this algorithm works by inserting Beijing followed by Shanghai into the front of a Deque. **(10 marks)**

(c) (15 marks)

Another variant of the basic Queue Abstract Data Type is a *Priority Queue*. Explain how this Priority Queue's work. Illustrate your answer by showing how the state of a sorted list-based implementation changes based on the following operations: insert(15, "mat"); insert(2, "the"); insert(8, "sat"), insert(7, "cat"); removeMin(); removeMin(); insert(12, "the"); removeMin(); insert(10, "on"); removeMin(); removeMin(); removeMin(). Show the state of the priority queue after each insert / remove operation.