



NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

SEMESTER 1

2005-2006

ALGORITHMS & DATA STRUCTURES 1
PAPER CS210

Dr. Philip Morrow, Prof. R. Reilly, Mr. T. Lysaght

Time allowed: 2 hours

Answer *three* questions

All questions carry equal marks

[25 marks]
[8marks]

- 1 (a) Write Java code to implement each of the following methods for a *stack* data structure which uses a static array. List the class instance variables necessary for these methods.

```

public ArrayStack (int size) // Constructor
public void Push(int S) //Add to the Top of
                        // the stack
public void Pop() //Remove the Top element of
                // the stack
public int Top() //Return the Top element of
                // the stack
private boolean IsEmpty()
private boolean IsFull()

```

- (b) Explain how a *stack* data structure may be implemented using the *ArrayList* data structure. Give pseudocode for the following methods with specific reference to the use of *ArrayList* methods: **[7 marks]**

```

public ArrayStack (int size) // Constructor
public void Push(int S) //Add to the Top of
                        // the stack
public void Pop() //Remove the Top element of
                // the stack

```

- (c) What significant difference is there between a data structure created with the use of an array and one created with a linked list? Explain how a stack data structure may be implemented using a linked list. **[10 marks]**

[25 marks]
[8 marks]

- 2 (a) Write Java code for each of the following methods for a *Queue Class*, *ArrayQueue*, to implement the operations of the *Queue* data structure using an array. List the class instance variables.

```
public ArrayQueue (int size) // Constructor
public void Enqueue(int S) //Add to the back
                        // of the queue
public void Dequeue() //remove from the front
                        //of the queue
public int getFront() // Return the front of
                        //the queue
private boolean IsEmpty()
private boolean IsFull()
```

- (b) Explain how the relevant methods from part (a) of this questions may be altered for the implementation of an array-based *altered-queue* of people. In the *altered-queue*, individuals are given a ticket number one greater than the person in front of them. Only the ticket number is stored in the array. Individuals are allowed to leave from the back of the queue as well as from the front. **[5 marks]**
- (c) Explain using diagrams, two methods of how a *queue* can be implemented as a circular array. Explain in pseudocode or Java code how the back and front references can be incremented in a circular array. **[7 marks]**
- (d) Briefly discuss the use of *interfaces* in Java. **[5 marks]**

- 3 (a) Give a definition for the Order-of-Magnitude of an algorithm (Big-O notation). Prove that the order of magnitude of the following function $f(n)$ is $O(n^2)$. [25 marks]
[8 marks]

$$f(n) = n^2 - 3n + 10$$

- (b) Write Java code to implement each of the following linked-list methods to add or remove or return the first element in the list. Write down the class instance variables necessary for the methods. [10 marks]

```
LinkedList() // Constructor
public Object getFirst()
public Object removeFirst()
public Object addFirst()
```

- (c) The linked list class in the Java Library supports doubly-linked lists including a reference, `precede` (opposite to `next`) to the predecessor node in the list. Use diagrams to show the changes to the links in a linked list under the `addFirst` and `removeFirst` methods that shows how the previous links need to be updated. [7 marks]

- 4 (a) Show the order in which the elements of the tree in Figure 1, would be processed for the following traversals: **[25 marks]**
[6 marks]

- (i) Pre-Order
- (ii) In-Order
- (iii) Post-Order

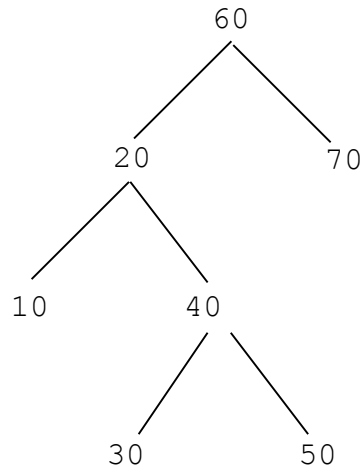


Figure 1.

- (b) The following elements are inserted into a Binary Search Tree. Make a drawing that shows the resulting tree after each insertion. **[8 marks]**

1, 10, 23, 20, 40, 5, 12

Explain, using diagrams, how a node with one child can be removed from a binary search tree. What is the difference between a balanced tree and an unbalanced tree?

- (c) Explain, using diagrams, how the *mergesort* recursive algorithm is implemented using a divide-and-conquer approach. **[6 marks]**
- (d) Discuss, in terms of big-O notation, the run time for the recursive *mergesort* algorithm. **[5 marks]**