



NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

**Second Computer Science And Arts Examination
Second Computer Science And Software Engineering Examination
Higher Diploma In Information Technology Examination
Master Of Computer Science (Year 1) Examination**

Year 2

**AUTUMN EXAMINATION
2004-2005**

**ALGORITHMS & DATA STRUCTURES 1
PAPER CS210/SE202A**

Dr. Philip Morrow, Prof. R. Reilly, Ms. S. Dunne

Time allowed: 2 hours

Answer *three* questions

All questions carry equal marks

[25 marks]
4 marks

- 1 (a) What, if anything, is wrong with the following loop?

```
int size = 10;
double[] data = new double[10];

for (int i = 1; i <= size; i++){
    data[i] = i * i;
}
```

[Note: There may be more than one error in the above piece of code.]

- (b) (i) What significant differences are there between a *static* data structure and a *dynamic* data structure? **6 marks**
- (ii) Give an example of when each of the above data structure types should be used.
- (c) Translate the following expressions into postfix notation. In your answer clearly explain the step-by-step process you use to carry out the translation. **9 marks**
- (i) $a - b - c + d * e$
- (ii) $a - b + (c + d) * e$
- (d) (i) Explain how the *height* of a tree affects the worst-case complexity of accessing any element in the tree. **6 marks**
- (ii) In your response compare the complexity of a '*skinny*' tree with the complexity of a '*complete*' tree.

[25 marks]
7 marks

- 2 (a) (i) Create a Binary Search Tree, entering these values in the order given:

56, 23, 10, 15, 77, 68, 4

(ii) What *levels* are the nodes 56 and 15 on?

- (b) Write a loop that reads five strings from the user and inserts them into an array. Write a second loop that prints the contents of the array in the opposite order from which they were entered. **5 marks**
- (c) Implement, using Java code, the *insertion sort* algorithm to sort an array of integers in **ascending** order. **8 marks**
- (d) Illustrate, using both text and using the diagram in Figure 1, each of the following traversal techniques: **5 marks**

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

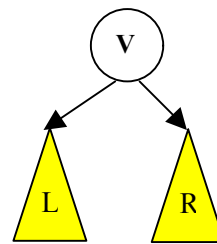


Figure 1

[25 marks]

- 3 (a) We express complexity using big-O notation. Examine the following loop that finds the minimum and maximum value in an array of length 'N'. What is the complexity for this loop?

7 marks

```
int min = MyArray[0];
int max = MyArray[0];

for (int i = 1; i < N; i++){
    if (min > MyArray[i]){
        min = MyArray[i];
    }
    else if (max < MyArray[i]){
        max = MyArray[i];
    }
}
```

- (b) In your answer book replace the ***** with code which will complete this QueueLinked class which implements a Queue ADT using a Linked List: **13 marks**

```
public class QueueLinked {

    public QueueLinked () {
        *****
        *****
        *****
    }

    public void enqueue(Object o) {
        *****
        *****
        *****
    }

    public Object dequeue()throws
    QueueEmptyException {
        *****
        *****
        *****
    }

    public Object peek () {
        *****
        *****
        *****
    }

    private ListNode front;
    private ListNode back;
```

```
private class ListNode{
    private ListNode next;
    private Object data;

    ListNode(Object o){
        data = o;
        next = null;
    }
    ListNode(Object o, ListNode n){
        data = o;
        next = n;
    }
}
```

[You may assume that `QueueEmptyException` has already been implemented.]

- (c) The use of a *current* reference is sometimes necessary in linked lists **5 marks** especially when deleting a listnode. Explain why it is necessary.

[25 marks]

- 4 (a) Describe the operation of the **Mergesort** algorithm when sorting an array. Clearly illustrate your answer with an appropriate example.

7 marks

- (b) *Compare* and *contrast* the **Stack** and the **Queue** data structures.

4 marks

- (c) In your answer book replace the ********* with code which will complete this StackArray class which implements a Stack ADT using an array.

10 marks

```

public class StackArray {

    public StackArray () {

        *****
        *****
        *****

    }

    public stack(int size){

        *****
        *****
        *****

    }

    public void push(Object o) throws StackFullException
    {

        *****
        *****
        *****

    }

    public Object pop() throws StackEmptyException{

        *****
        *****
        *****

    }

    public Object peek () throws StackEmptyException {

        *****
        *****
        *****

    }
    private Object []s;

    int size;

    int top;

}

```

[You may assume that StackEmptyException and StackFullException have already been implemented.]

- (d) Provide a detailed **pseudo-code** algorithm that uses one or more **stack** data structures to test if a mathematical expression inputted by the user has a matching number of *left* and *right* parentheses. **4 marks**

You may assume:

- (i) the input is in the form of a string
- (ii) only one type of parentheses '(' and ')' exists.