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**
**Masters Of Computer Science (Year 1) Examination**

**Year 2**

**SEMESTER 1**
**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

**1** (a) Explain how the *efficiency* of a method differs from the *correctness* of a method. **[3 marks]**

(b) Estimate the worst-case (big-O notation) time complexity in one call of the following method by counting the number of statements executed when the method is called once: **[10 marks]**

```
public int counting(int []a, double avg, int n){
    int counter = 0;
    for (int j = 0; j < N; j++){
        if(a[i] > avg){
             counter++;
        }
    }
    return counter;
}
```

(c) In your answer book replace the "********"s with code which will complete this StackLinked class which implements a Stack ADT using a Linked List: **[12 marks]**

```
public class StackLinked {

 public StackLinked () {
    ********
    ********
    ********
  }


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

  public Object pop(){
    ********
    ********
    ********
  }

  public Object peek ()  {
    ********
    ********
    ********
   }
  private ListNode top;
}
```

**2** (a) Compare and contrast the implementation of a data structure created with (i) the use of an array and, (ii) a linked list. **[4 marks[**

(b) Translate the following expressions into postfix notation: **[8 marks[**

(i) x + y – z / p * q
(ii) x + y – z * ( a / b )

(c) What would a QueueArray ADT, explicitly specified as an array of integers, contain if initially it was empty, and then had the following given sequence of operations performed on it? **[10 marks[**
Draw diagrams to show the contents of the Queue after each operation has been executed.

MyQueue.enqueue(2015);
MyQueue.enqueue(1200);
MyQueue.enqueue(1039);
MyQueue.dequeue();
MyQueue.enqueue(1181);
MyQueue.dequeue();
MyQueue.dequeue();
MyQueue.enqueue(1976);
MyQueue.dequeue();

(d) Describe how a Priority Queue differs from a Queue ADT. **[3 marks]**

**3** (a) Illustrate, using diagrams, how the array of integers in Figure 1 is sorted using: **[10 marks]**

(i) Insertion Sort
(ii) Selection Sort

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 25 | 17 | 33 | 9 | 2 |

Figure 1

(b) Write a method, `int sum(int[] MyArray)`, that returns the sum of the elements in the array MyArray. **[4 marks]**

(c) Show the order in which the elements of the tree in Figure 2, would be processed for the following traversals: **[6 marks]**

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

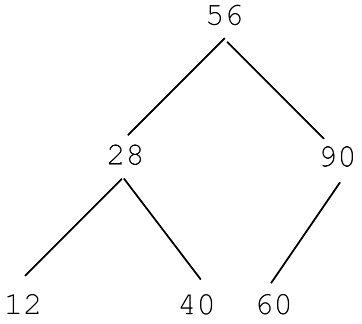

Figure 2

(d) Binary search can be said to offer **optimal worst cost**. Explain what this means. What is the complexity of binary search? **[5 marks]**

**4** (a) Create a Binary Search Tree, entering these values in the given sequence: **[5 marks]**

`15, 7, 3, 35, 44, 10, 8, 40`

(b) Calculate the height <u>of each node</u> in the tree you constructed in 4(a). **[5 marks]**

(c) ***Mergesort*** uses a divide and conquer technique to sort a sequence of input values. Implement a MergeSort Algorithm in Java to sort a randomly initialised array of doubles into **increasing** order. In your answer book replace the "********"s with code which will complete this sort. **[12 marks]**

```
/**Recursive algorithm to sort an array of integers
   @param array the array to be sorted
   @param start index position of the first element in the array region to
      be sorted.
   @param end index position of the last element in the array region to be
      sorted.
*/
 public static void mergeSort(int [ ] array, int start, int end){
    ********
    ********
    ********
  }


/**Merges two adjacent sorted regions of an array of integers
    subarray[start .. middle] and subarray[middle+1..end]

   @param array the array to be sorted
   @param start index position of the first element in sorted sub-array1.
   @param middle index position of the last element in sorted sub-array1.
   @param end index position of the last element in sorted sub-array2.
*/
public static void merge(int [ ] array, int start, int middle, int end){
    ********
    ********
    ********
}
```

(d) What is the benefit in using the Comparable data interface type, eg: **[3 marks]**

```
public Comparable data;
```