# NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

# OLLSCOIL NA hÉIREANN MÁ NUAD

# THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

## AUTUMN 2010 EXAMINATION

# CS210

# Algorithms & Data Structures 1

Dr. M. McNeill, Dr. A. Winstanley, Dr. P. Maguire

Time allowed: 2 hours

Answer *three* questions

**All questions** carry equal marks

**[25 marks]**

**1** (a) Compare and contrast the advantages and disadvantages of using **[6 marks]** ordered, unordered arrays and linked lists, with reference to insertion, deletion, searching and memory use.

(b) Describe, using suitable examples and diagrams, an algorithm for **[6 marks]** inserting values into an ordered array. Provide a Java implementation of the algorithm and include comments which explain your code.

(c) Describe, using suitable examples and diagrams, the binary **[5 marks]** search algorithm for ordered arrays. On average, how many steps would it take to search through an array with 100 elements using binary search?

(d) Describe, using suitable examples and diagrams, an algorithm **[4 marks]** which randomizes the ordering of elements in an array.

(e) Describe the Sieve of Eratosthenes algorithm for finding prime **[4 marks]** numbers within a given range, using examples and diagrams as appropriate.

**[25 marks]**

**2** (a) Explain the concept of Big O Notation in your own words, using **[4 marks]** examples and diagrams as appropriate. If an algorithm is O($n^n$), what would the implications be for running the algorithm?

(b) A function f($n$) describes the number of steps performed by an **[4 marks]** algorithm, where $n$ is the size of the problem and

$$\text{f}(n) = 9n + n\log n + 9$$

State the Big O complexity of the algorithm and prove that this is the case using the formal definition.

(c) Analyze the following segments of code and for both cases answer **[6 marks]** the following questions:

    i.    If $n$ is 0 how many times does each loop run?
    ii.   If $n$ is 5 how many times does each loop run?

iii. In light of this, derive an expression for the number of times each loop will run in terms of *n*

iv. State the Big O complexity of the code and explain your reasoning clearly

Case 1:

```
for(int i = n; i < n+10; i++){
   for(int j = n; j < 10n; j++){
      counter++;
   }
}
```

Case 2:

```
for(int i = 0; i <= 20; i++){
   counter++;
}
for(int j = 20; j > n; j--){
   counter++;
}
```

(d) Describe the following sorting algorithms: **[6 marks]**

    i. Bubble Sort
    ii. Selection Sort
    iii. Insertion Sort

Show how each algorithm would sort the following numbers, noting each intermediate arrangement:

      37    16    44    98    21    48    13

(e) Provide a Java implementation of the Bubble Sort algorithm. **[5 marks]**

**[25 marks]**
**3** (a) Describe the following data structures in detail, using examples **[9 marks]**
and diagrams as appropriate.

    i. Stack
    ii. Queue
    iii. Priority Queue

(b) Show how the data contents of a Stack, Queue and Priority Queue **[4 marks]**
would change given the following input and output commands.
Assume that bigger numbers have higher priority.

```
add (2)
remove ()
add (5)
add (9)
peek ()
add (1)
remove ()
remove ()
```

(c) Describe an algorithm which uses a stack to reverse a word. **[8 marks]**
Provide a full Java implementation of the algorithm, including both
the main class and the Stack class. Include comments which
explain your code.

(d) Show, using suitable examples and diagrams, how 'wraparound' **[4 marks]**
allows a queue data structure to be implemented using an array.
How can you tell when the queue is full?

**[25 marks]**
**4** (a) Explain in detail the concept of a linked list. Show how insertion **[5 marks]**
and deletion works for a linked list and provide a Java
implementation of a Link object.

(b) Using appropriate diagrams show the steps involved in inserting a **[4 marks]**
link at the head of a double-ended doubly-linked list.

(c) Describe the concept of recursion in your own words and identify **[3 marks]**
the key characteristics of a recursive method.

(d) What output would be returned by the following recursive method **[4 marks]**
given an input of 6? Explain your reasoning carefully.

```java
public int method(int number){

    if (number == 3){
        return 0;
    }
    return method(number - 1) + 1;
}
```

(e) Describe the mergesort algorithm in detail, using examples and diagrams as appropriate. **[6 marks]**

Show step by step how mergesort would sort the following set of numbers:

```
86   23   52   85   74   21   56   46
```

(f) Explain the following piece of code from the mergesort algorithm: **[3 marks]**

```
public void sort(long[] workSpace, int lower,int upper){
    if(lower == upper)
        return;
    else{
        int mid = (lower+upper) / 2;
        sort(workSpace, lower, mid);
        sort(workSpace, mid+1, upper);
        merge(workSpace, lower, mid+1, upper);
    }
}
```