OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

AUTUMN 2011 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) What is an algorithm? How was the concept of a computable algorithm formally defined by Alan Turing? [3 marks]

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

(c) Describe, using suitable examples and diagrams, the binary search algorithm for ordered arrays. Show the steps by which the binary search algorithm would find the number 79 in the ordered array given below: [5 marks]

```
6  11  32  38  51  60  67  69  76  79  92
```

(d) Describe, using suitable examples and diagrams, an algorithm which randomizes the ordering of elements in an array. Provide a Java implementation of the algorithm and include comments which explain your code. [6 marks]

(e) The following method is supposed to conduct a linear search on an array and return true or false depending on whether the input value is located or not. Rewrite the code without errors and include comments which explain the code. [5 marks]

```
public double search(searchKey){
   for(j=0; j<=array.size; j--){
       if(array[i] = key){
            j = key;
   }
   return false;
}
```

[25 marks]

2 (a) Explain in detail the concept of a linked list, using examples and diagrams as appropriate. Can a binary search be used on a linked list? Justify your reasoning. [5 marks]

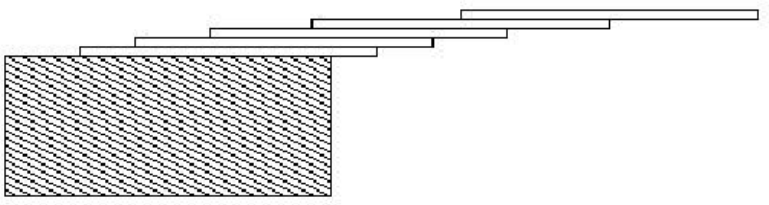
(b) Design and describe an algorithm that swaps the head and the tail of a single-ended doubly-linked list. Show step by step how this algorithm would manipulate the linked list, using an example to illustrate the process. [5 marks]

(c) Identify the errors in the following method for deleting a link with a particular value from a double-ended doubly linked list. The method should return the deleted node if it is found and return `null` otherwise. Rewrite the code without errors, adding [5 marks]

comments which explain what the code is doing.

```
public int delete (int key) {
     Link current = first;
     while(current != key){
        current = current.previous;
     }
     if(current = null){
        return null;
     }
     if(current = first){
        first = first.next;
     else
        current.previous = current.next;
     }if(current == last){
        last = previous;
     }else{
        next.previous=current.previous;
     }
     return current;
}
```

(d) How far can you make a stack of cards overhang a table? If you have one card, you can create a maximum overhang of half a card length. With two cards you can make them overhang by a total of three quarters of a card length. In general you can make n cards overhang by 1/2 of (1 + 1/2 + 1/3 + 1/4 + ... + 1/n) card lengths. This is illustrated in the figure below. [5 marks]



Write a Java program that outputs the minimum number of cards necessary to achieve a given overhang, which is input by the user. For example, if the user inputs “3.71” then the program should output “937 cards”, as this is the total number of cards needed to achieve an overhang of 3.71 card lengths.

(e) The ancient Greeks had a system for calculating square roots using addition and division. The algorithm is as follows: [5 marks]

1. Let *y* be your first guess for sqrt(*x*)
2. Now calculate (*y* + *x*/*y*)/2 to get a more accurate estimate

3. Keep applying the above process iteratively to refine your estimate

Write a Java program which reads in a number to square root from the user and then applies the above process ten times to produce an estimate (taking 1 as the initial guess).

[25 marks]

3 (a) Describe the concept of Big O Notation in your own words, using examples and diagrams as appropriate. Explain what the following notations mean: [5 marks]

1. O(1)
2. O(logn)
3. $O(n^3)$

(b) Your colleague claims that a program that took 100 seconds to run on 100KB of data will take 200 seconds to run on 200KB of data, because the algorithm is known to be O(n). Is this claim valid? Justify your reasoning. [3 marks]

(c) Analyse the following segment of code and answer the following questions: [4 marks]

i. If *n* is 2 what is the final value of counter?
ii. If *n* is 5 what is the final value of counter?
iii. In light of this, derive an expression for the number of times the inner loop will run in terms of *n*
iv. State the Big O complexity of the code and explain your reasoning clearly

```
int counter = 0;
for(int i = n; i <= 3*n; i++){
   while (i >= n){
      i--;
      counter++;
   }
}
```

(d) Write a Java method that uses recursion to compute a given term of the Fibonacci sequence. The first two numbers in the Fibonacci sequence are 0 and 1. Each subsequent number in the sequence is the sum of the previous two. Include comments in your code which explain clearly how the recursion works. [5 marks]

(e) Describe the mergesort algorithm in detail using examples and diagrams as appropriate. Show how mergesort would sort the numbers below. [8 marks]

65 32 97 14 47 73 53 25

[25 marks]

4 (a) Describe the following data structures in detail, using examples and diagrams as appropriate. [6 marks]

i. Stack
ii. Queue
iii. Priority Queue

(b) Show how the data contents of a Stack, Queue and Priority Queue would change given the following input and output commands. Assume that bigger numbers have higher priority. [3 marks]

```
 add (9)
 add (2)
 peek ()
remove ()
 add (3)
 add (6)
 add (1)
remove ()
 add (4)
 peek ()
remove ()
remove ()
```

(c) Explain how the concept of wraparound is applied so that an array can be used to represent a queue. Why do queues implemented using stacks not require wraparound? [4 marks]

(d) Describe in detail the following sorting algorithms: [12 marks]

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

As part of your explanation, show how each algorithm would sort the following numbers, noting each intermediate arrangement:

24 75 92 8 63 42 18 45