Durham University

EXAMINATION PAPER

Examination Session:	Year:	Exam Code:
May	2015	COMP1081WE01

Title: Algorithms and Data Structures

Time Allowed:	2 hours	
Additional Material provided:		
Materials Permitted:		
Calculators Permitted:	No	Models Permitted:
Visiting Students may use dictionaries: Yes		

Instructions to Candidates:	Answer FOUR questions. (TWO compulsory questions from Section A and TWO from Section B) ANSWER EACH SECTION IN A SEPARATE ANSWER BOOK

Revision:	

Section A Dr. Ioannis Ivrissimtzis and Dr. Matthew Johnson

Question 1

(a) Define the data structure **stack** and the associated operations **push** and **pop**. Give an example of when this data structure might be used. **[6 Marks]**

(b) What does an initially empty stack contain after the following sequence of operations?

push(4), push(8), pop, push(7), push(1), pop, push(6), pop, pop, push(9).

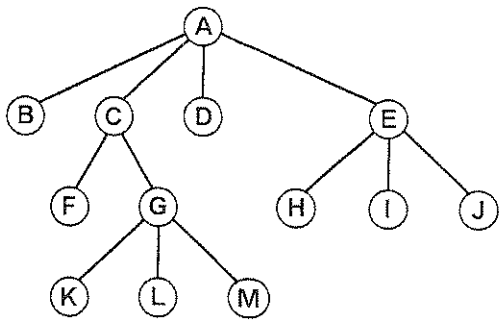
Moreover, what do the operations **top** and **isEmpty** return if applied after the sequence above? **[4 Marks]**

(c) Briefly explain how an array can be used to implement a stack, and write pseudocode for the push and pop operations. **[6 Marks]**

(d) Briefly explain how one array can be used to implement two stacks S_1 and S_2, and write pseudocode for the push and pop operations for each stack. The push operations should not return an error unless every cell in the array is used. **[9 Marks]**

Question 2

(a) Let T be the following tree:



i. What is the height of T?

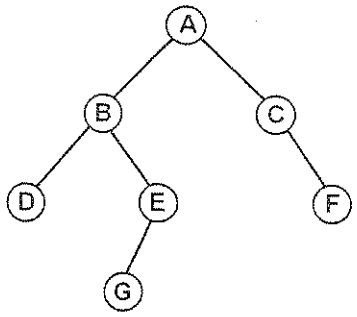
ii. How many internal nodes does T have?

iii. What is the depth of node B?

iv. Which nodes are in the subtree rooted at node E?

[4 Marks]

(b) Let T be the following binary tree:



Write down the list of letters that would be produced by a pre-order traversal of T.

[4 Marks]

(c) Describe the **linked structure implementation** of a binary tree and briefly discuss its advantages and disadvantages compared to the **array-list implementation**.

[5 Marks]

this question is continued on the next page

(d) Give a high level description of the **Merge Sort** algorithm for sorting a sequence S with n elements. **[4 Marks]**

(e) For initial input $S = \{ 41, 10, 29, 21, 7, 13, 46, 24 \}$ draw:

i. The merge sort tree T, with the input sequences processed at each node of T.

ii. The merge sort tree T, with the output sequences generated at each node of T.

[8 Marks]

End of Section A

Section B Dr. Ioannis Ivrissimtzis and Dr. Matthew Johnson

Question 3

(a) Let f and g be functions from the set of real numbers to the set of real numbers. We say that $f(x)$ is $O(g(x))$ if there are constants C and k such that

$$|f(x)| \leq C \cdot |g(x)|$$

whenever $x > k$.

Using the above definition, show that

$$(3x^2 + 4x + 1) \quad \text{is} \quad O(x^2).$$

[6 Marks]

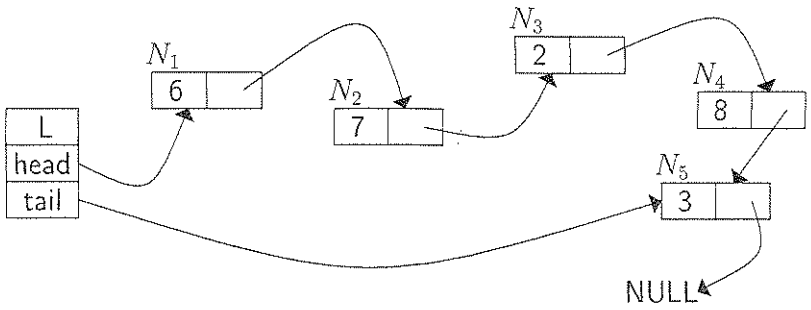
(b) Let f and g be functions from the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Omega(g(x))$ if there are constants C and k such that

$$|g(x)| \leq C \cdot |f(x)|$$

whenever $x > k$.

Using the above definition, show that if $f(x)$ is $\Omega(x^2)$ then $2 \cdot f(x)$ is also $\Omega(x^2)$. **[6 Marks]**

(c) Consider the singly linked list L illustrated below.



What do the following refer to?

i. L.head.next.data

ii. L.head.next.next.next **[4 Marks]**

this question is continued on the next page

(d) Give one advantage of using a linked list rather than an array. **[2 Marks]**

(e) Suppose that N is a node in a singly linked list that is known to *not* be the tail node. Describe by writing a short snippet of pseudocode how the piece of data contained in N can be deleted from the list. The time taken for this operation should be independent of the size of the list. Note that this implies you cannot refer to the node before N in the list. **[7 Marks]**

Question 4

(a) Give a high level description of the **Quick Select** algorithm. Briefly state the time complexity of the Quick Select algorithm, including the **best-case**, **average-case** and **worst-case** time complexity. **[6 Marks]**

(b) Insert, into an empty binary search tree, entries with keys 25, 35, 19, 53, 43 and 21 (in this order). Draw the tree after each insertion. **[6 Marks]**

(c) Define a **minimum spanning tree** (MST) of a connected undirected graph with weights on the edges. **[2 Marks]**

(d) Suppose that G is a connected undirected graph with weights on the edges. Show that if all the edges have distinct weights, there is a unique MST. **[4 Marks]**

(e) Let G be a connected undirected graph with distinct weights on the edges. An edge e of G is **dangerous** if it is the heaviest edge of some cycle in G. An edge e of G is **useful** if it is does not belong to any cycle of G.

i. Prove that the MST of G contains every useful edge.

ii. Prove that the MST of G does not contain any dangerous edge.

iii. Consider the following algorithm: the edges of G are considered in decreasing order of weight, and if an edge is dangerous it is removed. Prove that the resulting graph is the MST of G. **[7 Marks]**

Question 5

(a) Consider the Volta numbers V_n ($n \geq 1$ integer) defined by

$$V_n = \begin{cases} 1 & \text{for } n \leq 4, \\ V_{n-4} + 2V_{n-3} + 3V_{n-1} & \text{for } n \geq 5. \end{cases}$$

i. Calculate V_8. **[2 Marks]**

ii. Write pseudo-code for a **non-recursive** function that returns V_n for an integer $n \geq 1$. **[4 Marks]**

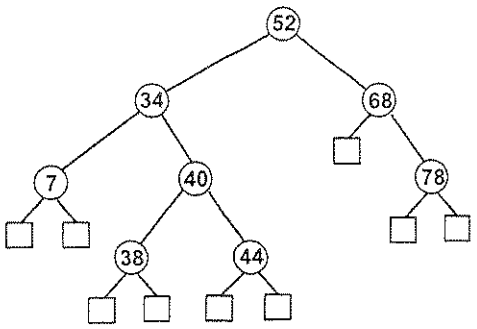
iii. Write pseudo-code for a **recursive** function that returns V_n for an integer $n \geq 1$. **[4 Marks]**

iv. How could you improve the running time of the recursive function? **[2 Marks]**

(b) Give the definitions of a **complete binary tree** and a **heap**. **[3 Marks]**

(c) Describe the **up-heap bubbling** process for inserting a new node in a heap. **[3 Marks]**

(d) Let T be the following AVL tree:



Draw the AVL tree resulting after the removal of the entry with key 52 from T. **[7 Marks]**

END OF PAPER