THE OPEN UNIVERSITY OF SRI LANKA
B.Sc DEGREE PROGRAMME: LEVEL 03
FINAL EXAMINATION: SEMESTER 2 - 2010/2011
CPU1142: DATA STRUCTURES AND ALGORITHMS

DURATION: **TWO HOURS** (2 HOURS)

Date: **29th June, 2011** Time: **1.30 pm – 3.30 pm**

Answer FOUR Questions ONLY.

Q1.

a)

i. Distinguish between *Abstract Data Types (ADT)* and *Data Structures*.

ii. Give a simple example for a Data Structure.

iii. Explain in your words the importance of having a Data Structure.

b) Define an ADT for character Strings. Your ADT should consist of at least four functions that can be performed on Strings.

c)

i. Briefly explain what is an *Algorithm*.

ii. Write a simple algorithm to solve the problem of finding the largest value in an array of *n* integers.

d) Explain the following terms briefly.

i. Best - case running time

ii. Average - case running time

iii. Worst - case running time

e) Determine the average case running time of the following *for* loop.

```
sum = 0;
for (i=1; i<=n; i++)
   sum += n;
```

Q2.

a)

i. Explain the *Array implementation of a linked list* and *Pointer implementation of a linked list* in your own words.

ii. What is the advantage of a *Pointer implementation of a linked list* over an *Array implementation of a linked list*?

b) Consider the following figure which illustrates a portion of the array implementation of a linked list named *linkArray* that contains three linked lists. List 1 starts at *linkArray[1]*, list 2 starts at *linkArray[7]* and list 3 starts at *linkArray[17]*. Write down the order of the values which are stored in the lists separately.

	info	Next
0	55	15
list 1 = 1	25	4
2	54	-1
3	52	10
4	16	0
5		
6		
list2 = 7	59	21
8	72	19
9		
10	11	-1
11		
12		
13		
14		
15	20	-1
16		
list3 = 17	12	8
18		
19	66	2
20		
21	6	3
22		

c) Write C program codes to do the following operations on a Pointer implementation of linked lists.

i. To insert a node at the beginning of the linked list.

ii. To insert a node at the end of the linked list.

iii. To delete the node at the end of the linked list.

Q3.

a) What is a *Stack*? Explain using an example.

b) Differentiate between the *Array implementation of a Stack* and *Pointer implementation of a Stack.*

c) Write C program codes for the following basic operations that can be performed using an Array implementation of a Stack.

`PUSH(X,S)`	–	This operation inserts the element X at the top of the stack S. The old top element becomes the next-to-top element and so on.
`POP(S)`	–	Delete the top element of the stack.
`TOP(S)`	–	Return the element at the top of the stack without deleting from the stack, S.
`EMPTY(S)`	–	Return true if S is an empty stack. Otherwise return false.

d) Consider the Pointer implementation of a Stack for answering the following parts (i) and (ii).

i. Illustrate a Stack by inserting the following set of integers respectively.

35, 40, 50, 10, 12

ii. Write down the C program code to insert an element into a Stack.

Q4.

a) A graph consists of a set of objects (called vertices) and a set of edges, where each edge connects two vertices. Any given pair of vertices can be connected by only one edge. Describe two different ways to represent the connections defined by the vertices and edges of a graph.

b) Describe the following terms with regard to a graph.

i. Path

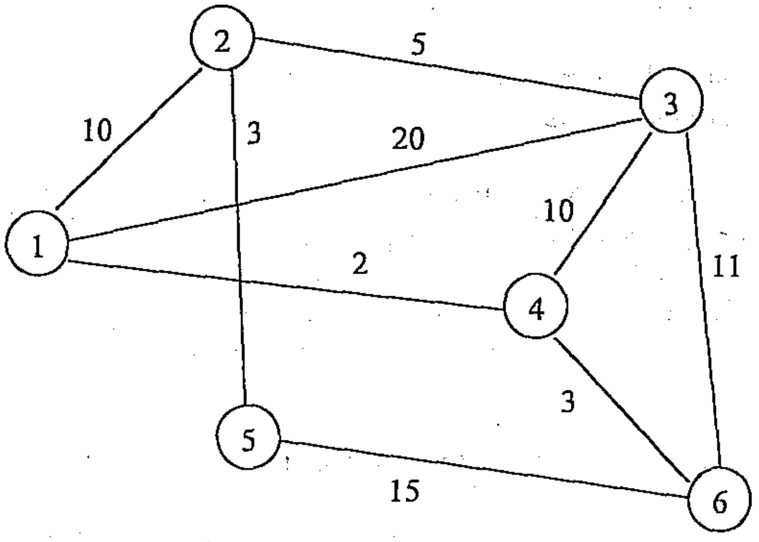
ii. Cycle

iii. Connected graph

iv. Complete graph

v. Weighted graph

c) Consider the following graph and answer the parts (i) and (ii).



i. Represent the above graph using the representations you described in part (a).

ii. Using any of the representations you described in part (a), apply the depth-first search to the graph and list the vertices they would be visited starting from node 1.

Q5.

a) Explain the following trees with their properties. Give one example for each.

i. Strictly binary tree

ii. Complete binary tree

iii. Almost complete binary tree

b) What is a binary search tree?

c) Construct a binary search tree for the following set of integers.

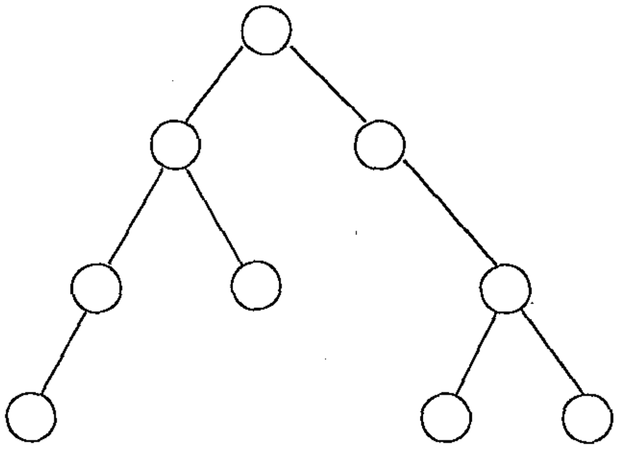
37, 24, 42, 7, 2, 40, 42, 32, 120

d) Consider the binary search tree constructed in part (c) and draw the resulting tree if you delete the node with the value 37.

e) What is an AVL tree?

f) What is the balance factor of the AVL tree?

g) Consider the following figure of an AVL tree. Show the balance factor of each node and indicate it within the node.



Q6.

a) Write down the C program code for the linear insertion sort algorithm.

b) Illustrate the insertion sort algorithm for sorting the following array with eight elements.

42	20	17	13	28	14	23	15

c) What is the efficiency of the linear insertion sort?

d) Define max-heap and min-heap with a simple example.

e) Draw an initial ascending heap for the following set of integers.

73, 6, 57, 88, 60, 42, 83, 72, 48, 85

*** All Rights Reserved ***