Comprehensive Examination Marking/Answering Scheme

BITS Pilani, Dubai Campus, Dubai International Academic City

III Year First Semester 2012-2013

Degree: B.E.(Hons.) Branch: C.S.

Course No : CS C363 Course Title: Data Structures and Algorithms [Theory]

Date: 31, Dec., 2012 Monday Time: 3 hours Total marks: 80

Weightage: 40% Data provided are complete. ***Closed Book***.

1.Define the following terms w.r.t. algorithms:

a)ANALYSIS b) DESIGN [2 M]

Analysis deals with time and space complexity. Design deals with design techniques like D&C, Greedy Method and so on.

2.What is a Minimum Spanning Tree? [2 M]

A tree constructed from undirected graph; whose sum of edges gives minimum cost.

3.Distinguish between Adjacency Matrix and Adjacency List in Graph ADT. [2 M]

Matrix shows the edges between neighbouring nodes in a 2 D array. List shows the linked edges of neighbouring nodes for each node.

4. Explain the meaning of the following methods w.r.t ITERATOR ADT:

c) **hasNext** b) **nextObject** [2+2 M]

test if there are elements left in the iterator. Return and remove the next element in the iterator.

5.a) **Trace** through all the steps for PATTERN MATCHING using BOYER-MOORE algorithm [algorithm need not be written; only tracing needed] for the following input data:

String **T**: COMPREHENSIVEEXAMINATIONIIIYEARCSSTUDENTSCALCULAT

Pattern **P**: RSCA

Trace with Looking Glass Heuristics and Character Jump Heuristics [5 M]

5.b) Explain the following operation in AVL tree with an example:

DOUBLE LEFT ROTATION

DLR=SLR+SRR example 1.5+1.5+2 M

6.Write a brief technical note on Insertion and Query operations in a BLOOM FILTER with an example scenario. INSERT + QUERY + SCENARIO: 3*2= M

7. **Solve** the following instance of SUM OF SUBSETS problem using BACKTRACKING (algorithm not needed; it is enough if you draw the state space tree as required):

n=7 (no of items)
m=32 (sum)
w = {5, 7, 10, 12, 15, 18, 20} (weights of individual items) State Space tree and elements and calculations in each branch. [10*1=10 M]
8. Huffman Coding Algorithm for text compression: Input, Logic, Output:1.5+7+1.5 M

9.Show (**trace through**) the successive steps of **HEAPSORT** for sorting the elements in ascending order for the array given below:
A = [47, 30, 26, 22, 23, 32, 54, 25]
Build Heap: 3 M
steps in Swap & heapify (array contents to be shown progressively) 7 M

10.Trace through the successive steps for inserting the following keys**, in the given order**, into an initially empty **B-Tree** of degree 3. (i.e. t=3; an internal node can have between 2 to 5 keys):
67, 39, 49, 88, 26, 50, 37, 57, 47, 91, 24, 31, 84, 51, 74, 61, 94, 21, 84, 11
[20*0.5=10 M]

11.Design an ALGORITHM that prompts the user for a number, then determines whether the number is ***perfect, deficient,*** or ***abundant***. Your algorithm should continue to prompt the user for numbers until a 0 is provided as input.
INPUT+LOGIC+OUTPUT [1+5+1=7 M]

12.Write an algorithm to insert elements in a SKIP LIST. I/L/O [1+5+1 M]

TEST II Question Paper

BITS, Pilani – Dubai Campus, Academic City, Dubai.

III Year FIRST SEMESTER 2012-2013

Degree: B.E. (Hons.) Branch: C.S.

Course No : CS C363 Course Title: Data Structures and Algorithms

Date: 18, November, 2012 Sunday Time: 50 min. Total marks: 40 Weightage: 20%

Data provided are complete. **OPEN *Book***.

Text / Reference Book and hand written class notes permitted.

This question paper has 2 pages.

Answer all Questions.

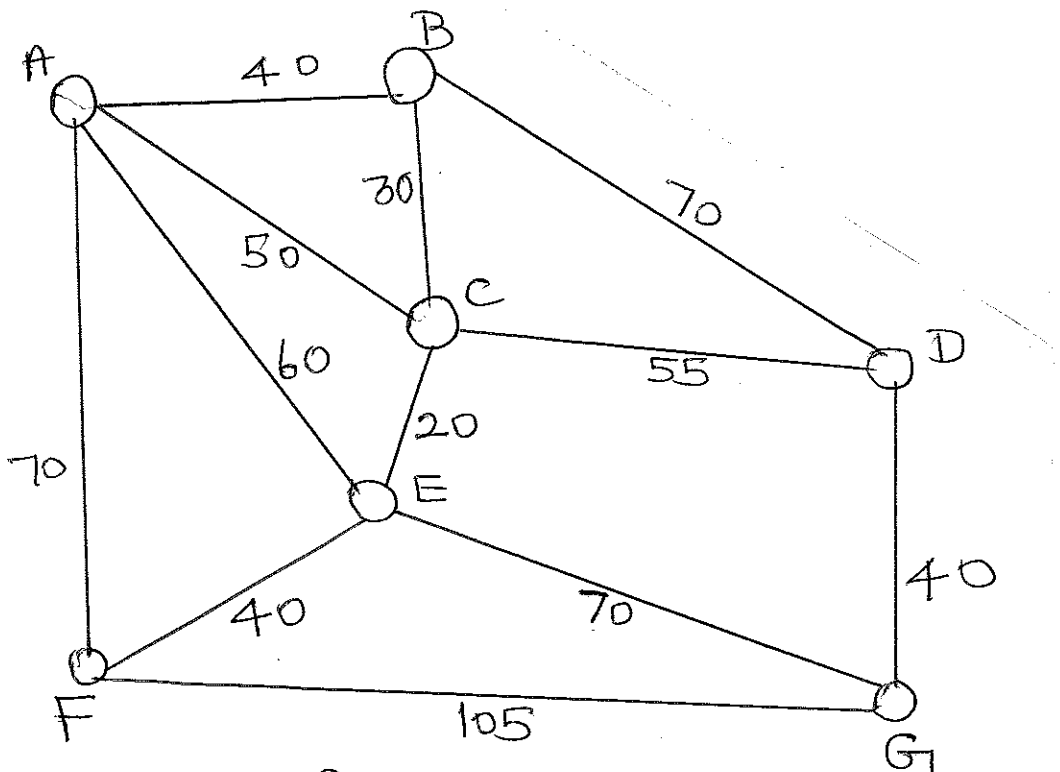
1. Consider the Adjacency Matrix for a **directed graph** G shown below:

	A	B	C	D	E	F	G	H
A	0	1	1	0	0	0	0	0
B	0	0	0	0	1	0	0	0
C	0	1	0	0	0	0	0	0
D	1	0	0	0	1	0	0	0
E	0	0	0	0	0	0	1	0
F	0	1	1	0	1	0	1	0
G	0	0	0	0	0	0	0	0
H	0	0	0	0	1	0	1	0

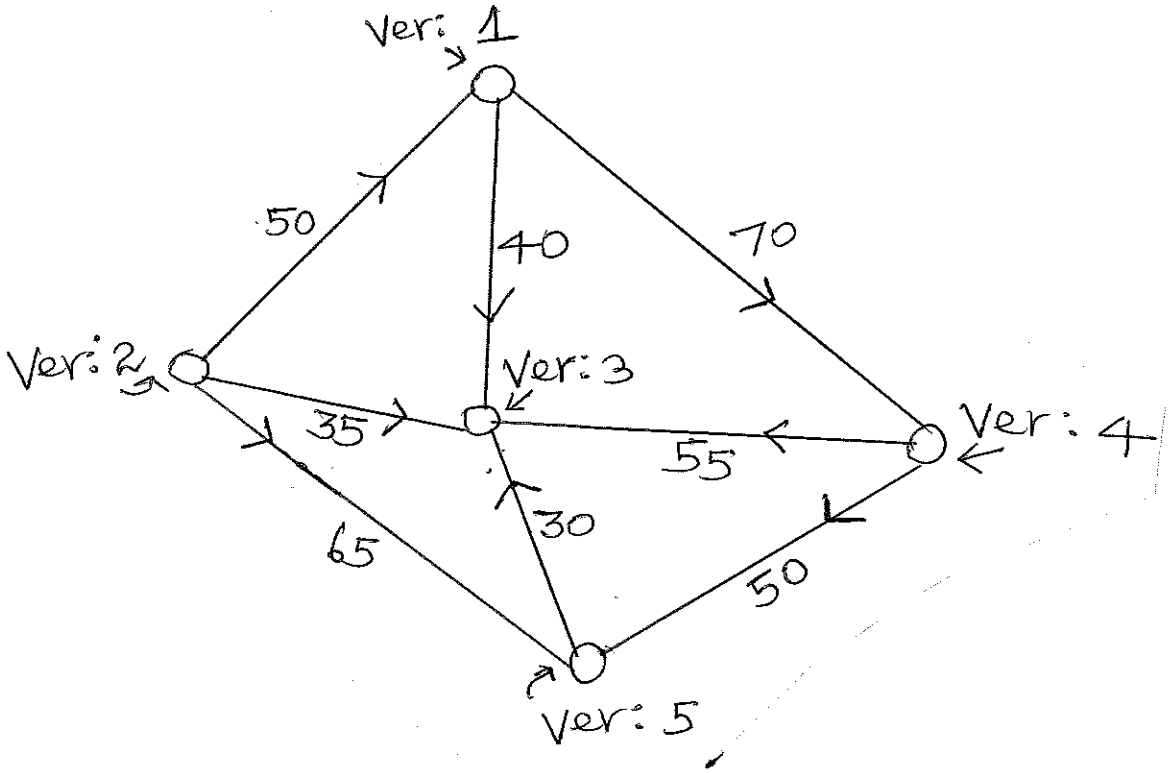
a) Draw the Graph G. [2 M]

b) Using DFS (show all steps), find all nodes that are **reachable** from node A and **Print** them. (Algorithm not need not be written) [5 M]

2. Construct the Minimum Spanning Tree for the weighted **undirected graph** shown below using PRIM's Algorithm.(algorithm need not be written). [8 M]



3. Solve the following **Single Source Shortest Paths Problem instance using Dijkstra's Algorithm** (do not write algorithm; just solve numerically)
Given a **directed graph** G(V, E), Determine the cost of the shortest path from the source to every other Vertex in V, where the length of a path is just the sum of the costs of the arcs on the path.
Here, you have compute the cost of the shortest path from **vertex 1** to all other vertices [i.e.**2, 3, 4, 5**] by showing all the steps. [8 M]



4. The following character string (given in next line) is to be transmitted using HUFFMAN CODING:

SINGLESOURCESHORTESTPATH

a) ***Construct*** the HUFFMAN Coding **Tree** for the letters present in the above string
b) ***Determine*** the number of bits required to code each letter. [9 M]

5. Find the **minimum number of multiplications** required for the MATRIX Chain-Product $A = A_0 \times A_1 \times A_2 \times A_3$ step by step, using **Dynamic Programming**. The dimensions of the Matrices are given as follows:

$A_0 = (7x5)$ $A_1 = (5x6)$ $A_2 = (6x9)$ $A_3 = (9x12)$

[8 M]

BITS Pilani, Dubai Campus, Academic City, Dubai.

III Year FIRST SEMESTER 2012-2013

Degree: B.E. (Hons.) Branch: C.S.

TEST I Question Paper

Course No : CS C363 Course Title: Data Structures and Algorithms

Date: 30, September, 2012 Sunday Time: 50 min. Total marks: 20

Data provided are complete. **Closed Book.**

This question paper has 1 page.

Answer all Questions.

1. You are required to calculate the growth (computing time in number of steps as a function of input size) for each of the given *functions.* Consider *only base 2* for logarithms. Fill up the following blank table entries:

Growth of f(n)

n (input size)	f(n) = n (log n)	f(n) = $2n+1+n^2$	f(n)= (log n) + n+1	f(n) = n^2
32				

[2 M]

2. Write the algorithm for computing the **depth** of a node **v** in a tree T. [2 M]
3. State formally the *BIG – O* notation w.r.t. time complexity in algorithmic studies. [2M]
4. Write down the time complexity for each of the following methods: [2 M]
   a) PUSH(o) in Stack (for array based implementation)
   b) POP(0 in Stack (for array based implementation)
   c) ENQUEUE(o) in QUEUE (for array based implementation and viewed circularly)
   d) DEQUEUE() in Queue (for array based implementation and viewed circularly)

5. Write an algorithm to implement the method **insertBefore(p,e)** in a doubly linked list. (i.e. insert a new element **e** into a doubly linked list S, before position **p** in S) [5 M]

6. Trace through (algorithm not to be written) the successive passes of MERGESORT for sorting the following input data:
   169, 140, 251, 332, 463, 594, 667, 117, 287, 228, 154, 150, 182, 136, 109, 207, 307, 402, 312, 297. [4 M]
7. Solve the following recurrence equation for the following problem:

T (n) =1, if n=1
T (n) = 2T(n-1) otherwise. [3 M]

BIT Pilani, Dubai Campus Academic City, Dubai.
III Year First Semester 2012-2013
Degree: B.E. Hons. Branch: C.S.
QUIZ I [SET A]
Course No : CS C363 Course Title: Data Structures and Algorithms
Date: 23, Oct., 2012 Tuesday Time: 20 min. Total marks: = 10
Weightage: 5% Venue : seating arrangement ***Closed Book***.
This question paper has 2 pages [back side: rough work]

IDNO: **Name:**

SET A

Write answers in the space provided in question paper. Answer all questions.

1. Draw a rough sketch of ZIG ZIG operation in a SPLAY TREE (showing before and after splay operation). [2 M]

2. Given input { 663, 223, 271, 351, 454, 306 } and a hash function h(X)= X mod 13, show the resulting ***Separate chaining hash table***.

[2 M]

BIT Pilani, Dubai Campus Academic City, Dubai.
III Year First Semester 2012-2013
Degree: B.E. Hons. Branch: C.S.
QUIZ I [SET A]
Course No : CS C363 Course Title: Data Structures and Algorithms
Date: 23, Oct., 2012 Tuesday Time: 20 min. Total marks: = 10
Weightage: 5% Venue : seating arrangement ***Closed Book***.
This question paper has 2 pages [back side: rough work]

IDNO: **Name:**

SET A

1. A) Binary Search Tree (BST)

Insert items with the following keys (in the given order, read from left to right) into an initially empty Binary Search Tree.
82, 15, 32, 13, 14, 26, 37, 24 [1.5 M]

3. B) For a BST with height h for n keys-element items, Operation **findElement** takes O( ) time and **isEMPTY** takes O( ) time. [1 M]

4. A) Define GREEDY- CHOICE property.

2 B) What is the worst case time complexity of Fractional Knapsack algorithm? [1+1 M]

5. Write the formal definition for SKIP LIST. [1.5 M]