

(3 Hours)

[Marks: 80]

N.B.: 1) Question No. 1 is compulsory.

2) Answer any three out of remaining questions.

3) Assume suitable data if necessary.

4) Figures to the right indicate full marks.

- Q1. (a) Explain different types of link list. (5)
 (b) List down the applications of stack. (5)
 (c) Explain directed and undirected graph with the help of example. (5)
 (d) Explain different hash functions. (5)
- Q2. (a) Define B-tree. Step by step construct a B-tree for the following data 23, 12, 25, 01, 45, 63, 27, 29 for order 3. (10)
- Q2. (b) Write an algorithm to perform following operations on singly link list: (10)
 i) Insertion from the Beginning
 ii) Deletion from End
 iii) Display the contents of list
- Q3. (a) Write an algorithm to covert infix expression to postfix expression. Convert the following infix expression to postfix expression using stack- (10)
 $A+B*(C-D)/(P-R)$
- Q3.(b) Consider a file containing 6 unique characters and frequency of each character is given:
 $c=34 \ d=9 \ g=35 \ u=2 \ m=2 \ a=100$
 How many bits are required to store this file using Huffman Encoding? (10)
- Q4. (a) Define Minimum Spanning Tree. Construct a minimum spanning tree shown in figure 1 using Kruskal's and Prim's Algorithm and find out the cost with all intermediate steps. (10)

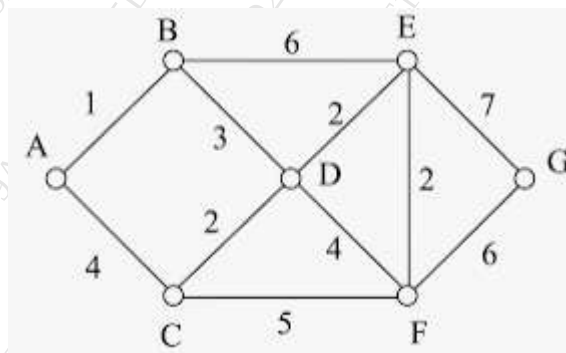


Figure 1

Q4.(b) Define AVL tree. Construct an AVL tree from the following data and mention the rotations in each step. (10)

40,30,20,25,21,50,60,70,65,22,18,15

Q5. (a) What is collision? List down the methods to resolve the collision. Consider a hash table of size 11. Using linear probing, insert keys 54, 26, 93, 17, 77, 60 and 31 into the table. (10)

Q5. (b) Define Binary Search Tree. Construct the binary search tree from following data and traverse the tree in in-order, pre-order and post-order traversal by giving its algorithm.

14, 15, 4, 9, 7, 18, 3, 5,16,20,17 (10)

Q6. Solve **any Four**: (20)

- a) Threaded Binary Tree
- b) Breadth First Search
- c) B-tree
- d) Graph Representation Techniques
- e) Game Tree
