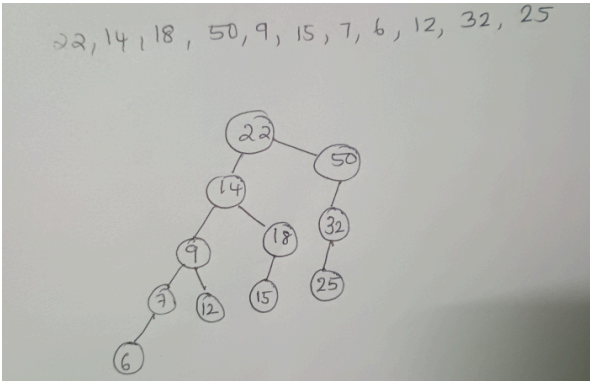


Sub:	DATA STRUCTURES AND APPLICATIONS				Sub Code:	BCS304
------	----------------------------------	--	--	--	-----------	--------

Date:		Duration:	90 minutes	Max Marks:	50	Sem/Sec: III A,B,C
-------	--	-----------	------------	------------	----	-----------------------

Scheme and Solutions

1	a	<p>Construct a binary search tree for the given values 14, 15, 7, 9, 18, 3, 5, 16, 20. Write the C function for the inorder, preorder and postorder traversal and apply the same.</p> <p><b>Answer:</b></p> <p><b>Construction of BST-4M(step wise)</b></p>  <p><b>Search an item in the BST-2M</b></p>
---	---	---

```

struct node * search ( struct node * root, int key )
{
    if ( root == NULL )
        return root;

    else if ( key == root->data )
        return root;

    else if ( key < root->data )
        search ( root->left, key )

    else if ( key > root->data )
        search ( root->right, key )
}

```

Explain winner tree and loser tree with suitable examples.

**Answer:**

b **Winner Tree Explanation with example-2M**

**Loser Tree Explanation with example-2M**

Construct a binary tree by using the following in-order and pre-order traversal.

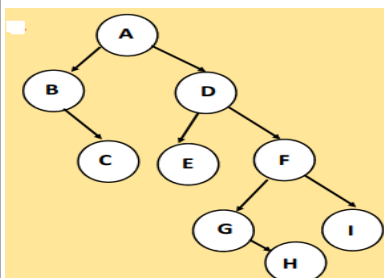
In-order: BCAEDGHI

Pre-order: ABCDEFGHI

Also perform the post order traversal of the tree.

**Answer:**

2 a **Construction of Binary Tree-3M**



### Postorder-1M CBEHGIFDA

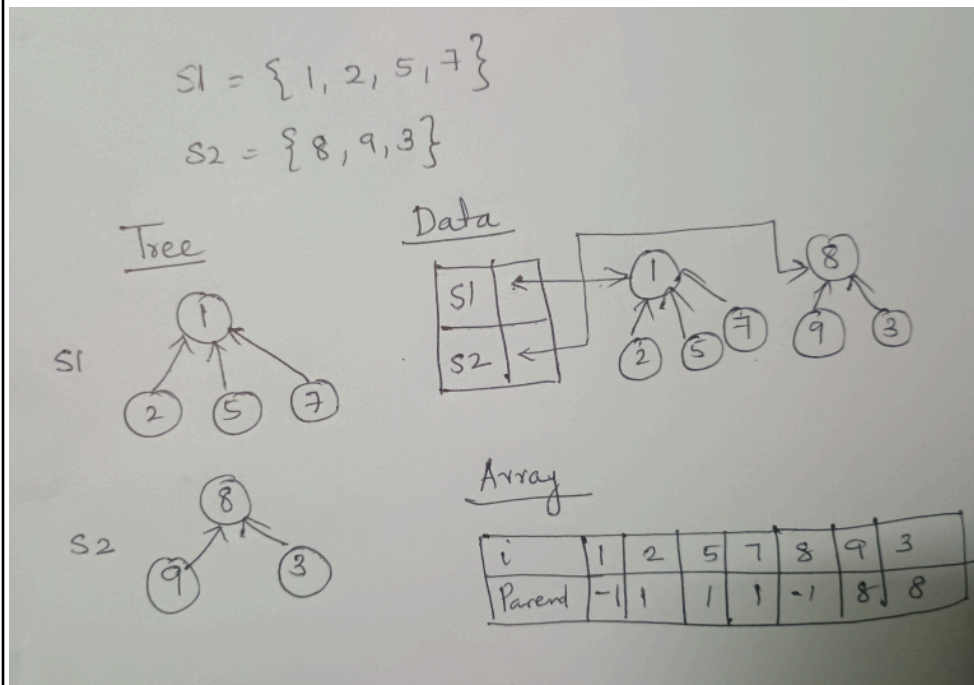
Demonstrate the tree, data, and array representation for the disjoint sets,

$S_1 = \{1, 2, 5, 7\}$

$S_2 = \{8, 9, 3\}$ . Also write algorithm for simple union () and simple find().

**Answer:**

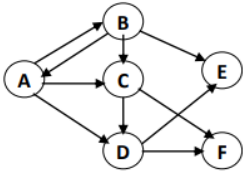
**Representation of tree, data, array -1M,2M,1M**

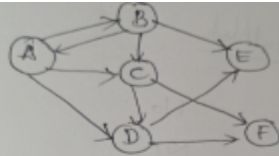


**simple union()- 1M**

```
void simpleUnion(int i, int j)
{
    Parent[i] = j;
}
```

**simple find()- 1M**

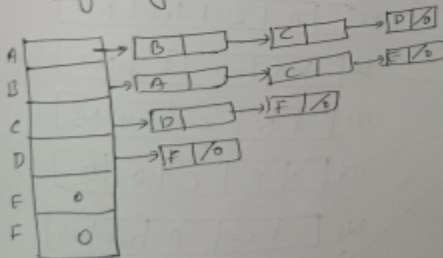
		<pre> find (c) {     while (P[u] ≥ 0)     {         i = P[u];     }     return i; } </pre>
3	a	<p>Define Graph. For the given graph, show all the three representations of the graph.</p>  <p><b>Answer:</b></p> <p><b>Representation of graph</b></p> <p><b>Adjacency Matrix 1M</b></p> <p><b>Adjacency List 2M</b></p> <p><b>Adjacency Multilist 2M</b></p>



### Adjacency Matrix

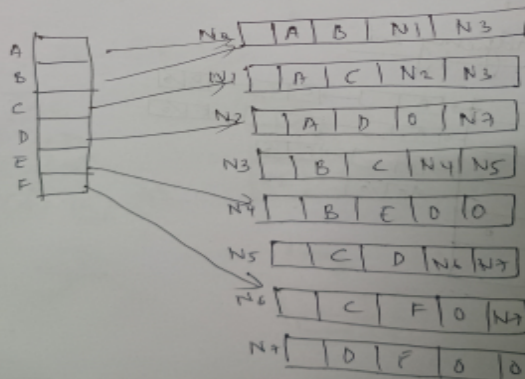
	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	0	1	0
C	0	0	0	1	0	1
D	0	0	0	0	0	1
E	0	0	0	0	0	0
F	0	0	0	0	0	0

### Adjacency List



### Adjacency Multilist

A, B	N <sub>0</sub>
A, C	N <sub>1</sub>
A, D	N <sub>2</sub>
B, A	N <sub>3</sub>
B, C	N <sub>4</sub>
B, E	N <sub>5</sub>
C, D	N <sub>6</sub>
C, F	N <sub>7</sub>
D, F	N <sub>8</sub>



What are the methods used for traversing a graph? Explain any one with example and write C function for the same.

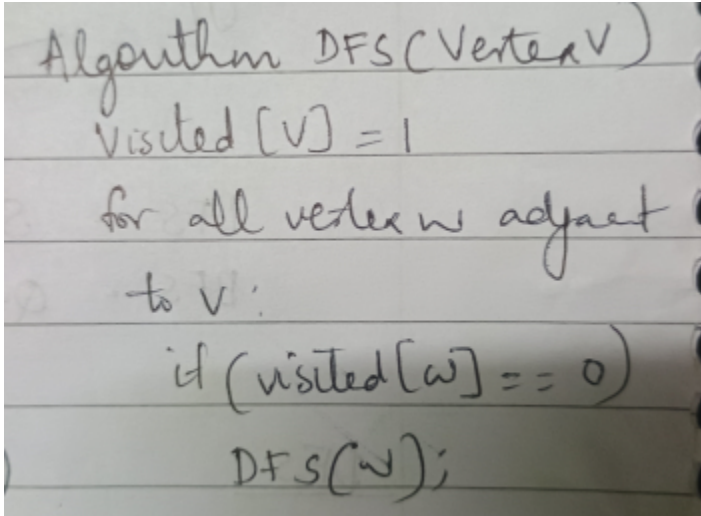
**Answer:**

**Methods for Traversing — 1M**

**DFS (Depth First Search) BFS (Breadth First Search)**

**Algorithm for DFS or BFS 2M**

b



```
Algorithm DFS(Vertex v)
visited[v] = 1
for all vertex w adjacent
to v:
    if (visited[w] == 0)
        DFS(w);
```

```

Algorithm BFS(V)
{
    A BFS of  $G(V, E)$  is carried out
    beginning at vertex V and array visited
    of n initially set to false

    visited[V] = true;
    initialise Queue(Q);
    add(Q, V);
    while (not empty Queue(Q)) do
    {
        v = delete(Q, V);
        for all vertex  $w$  adjacent to v
        {
            if not visited[w] then
            {
                add(Q, w);
                visited[w] = true;
            }
        }
    }
}

```

### Example 2M

Given a hash table with 10 slots. The hash function is  $h(k) = k \bmod 10$ . The collision is overcome by chaining. The following keys are inserted in the order. 5, 28, 19, 15, 20, 33, 12, 17, 10. Develop the corresponding hash table.

**Answer:**

a

4

Explain the following by taking suitable examples,  
a) Linear Probing b) Quadratic Probing c) Folding Method

**Answer:**

b

**Linear Probing technique with example-2M**

**Quadratic Probing technique with example-2M**

		<b>Folding technique with example- 2M</b>
5	a	<p>Explain dynamic hashing using directories with the help of an example.  <b>Answer:</b></p> <p><b>Dynamic hashing using Directories 2M</b></p> <p>-elements: 16,4,6,22,24,10,31,7,9,20,26.  Bucket Size: 3</p> <p>16- 10000</p> <p>4- 00100</p> <p>6- 00110</p> <p>22- 10110</p> <p>24- 11000</p> <p>10- 01010</p> <p>31- 11111</p> <p>7- 00111</p> <p>9- 01001</p> <p>20- 10100</p> <p>26- 11010</p>
	b	<p>Differentiate between height biased and weight biased leftist tree with examples.  <b>Answer:</b></p>



Height biased leftist tree 2.5M  
Weight biased leftist tree 2.5M

What is the need for an optimal BST. Find the optimal BST for  $n=4$ ,  
Keys are 10,15,20, 25.  
 $p_1, p_2, p_3, p_4 = 3,3,1,1$   
 $q_0, q_1, q_2, q_3, q_4 = 2,3,1,1,1$

**Answer:**

**Need for BST-2M**

**Problem-8M**

$W(i, i) = q(i)$   
 $C(i, i) = 0$   
 $\lambda(i, i) = 0$

	0	1	2	3	4
0	$W_{00} = 2$ $C_{00} = 0$ $\lambda_{00} = 0$	$W_{11} = 3$ $C_{11} = 0$ $\lambda_{11} = 0$	$W_{22} = 1$ $C_{22} = 0$ $\lambda_{22} = 0$	$W_{33} = 1$ $C_{33} = 0$ $\lambda_{33} = 0$	$W_{44} = 1$ $C_{44} = 0$ $\lambda_{44} = 0$
1	$W_{01} = 8$ $C_{01} = 8$ $\lambda_{01} = 1$	$W_{12} = 7$ $C_{12} = 7$ $\lambda_{12} = 2$	$W_{23} = 3$ $C_{23} = 3$ $\lambda_{23} = 3$	$W_{34} = 3$ $C_{34} = 3$ $\lambda_{34} = 4$	
2	$W_{02} = 12$ $C_{02} = 19$ $\lambda_{02} = 1$	$W_{13} = 9$ $C_{13} = 12$ $\lambda_{13} = 2$	$W_{24} = 5$ $C_{24} = 8$ $\lambda_{24} = 3$		
3	$W_{03} = 14$ $C_{03} = 25$ $\lambda_{03} = 2$	$W_{14} = 11$ $C_{14} = 19$ $\lambda_{14} = 2$			
4	$W_{04} = 16$ $C_{04} = 32$ $\lambda_{04} = 2$				

