Jaypee Institute of Information Technology
Test-2 Examination 2020-21

Course Name: Data Structures & Algorithms **Course Code: 15B11CI518**
Max. Time: 1 Hr **Max. Marks: 20**

Q.1 [C311.2] [1M] How many number of comparisons are required if interpolation search is applied to search value 20 in following array: 10, 12, 13, 16, 18, 19, 20, 21, 22, 23, 24, 33, 35, 42, 47

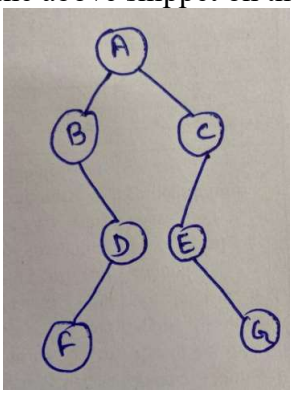
Answer: **4 comparisons**

lo	hi	arr[lo]	arr[hi]	pos	arr[pos]	Comparison
0	14	10	47	3	16	1
4	14	18	47	4	18	2
5	14	19	47	5	19	3
6	14	20	47	6	20	4

Q.2 [C311.1] [1M] Consider the following code fragment and assume no syntax error.

```
do
{
        if(t)
        {
                printf(t-> data);
                do(t -> left);
                printf(t-> data);
                do(t -> right);
        }
}
```

Apply the above snippet on the tree below and identify the output.



Answer: A B B D F F D A C E E G G C

Q.3 [C311.1] [1M]

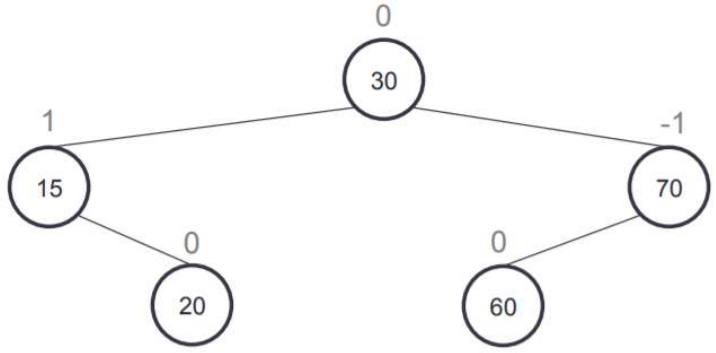
```
int findElement(int A[], int n)
{
        int Element = A[n / 2];
        for (int i = 1 + n / 2; i < n; ++i)
                Element = Element >A[i]? Element:A[i];
        return Element;
}
```

The above code is run on an array A which is a min heap tree. What will the code do?

Answer: Find the maximum element in the min-heap.

Q.4 [C311.1] [1M]

Consider the AVL tree shown below:



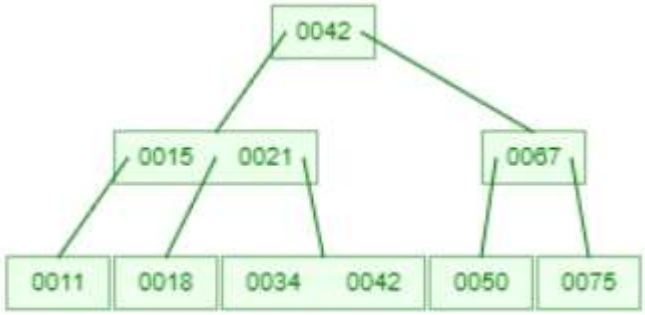
Inserting which of the following number(s) will result in single left rotation: 91, 5, 22, 80.

Answer: 22

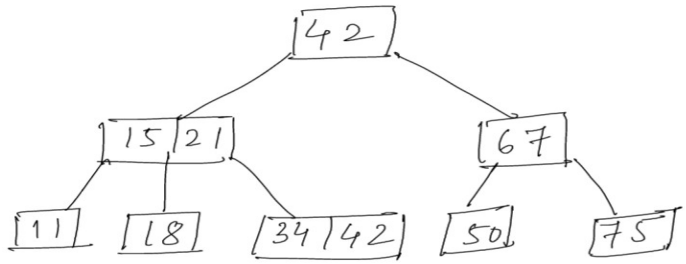
Q.5 [C311.1] [1M] Assume the nodes in a binary tree having no siblings are considered as lonely nodes. If there are a total of *n* nodes in an AVL tree, then what can be the maximum number of lonely nodes in this AVL tree?

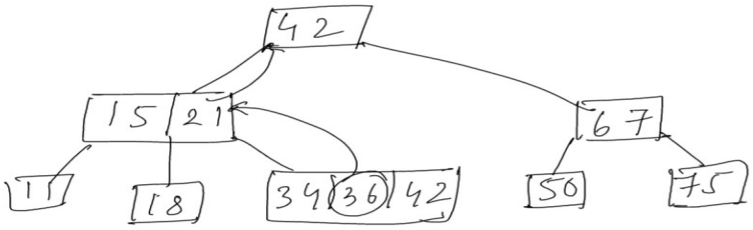
Answer: Less than equal to n/2.

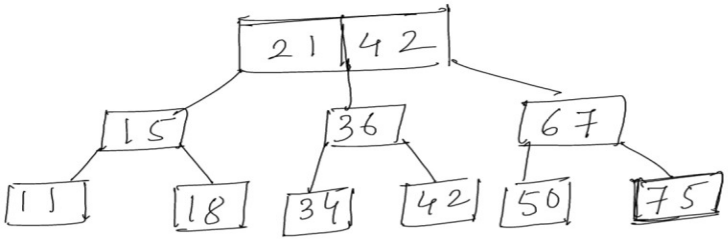
Q.6 [C311.1] [1M] A B-tree of order 3 has been shown below, perform insertion of 36 in this B Tree.



Answer:







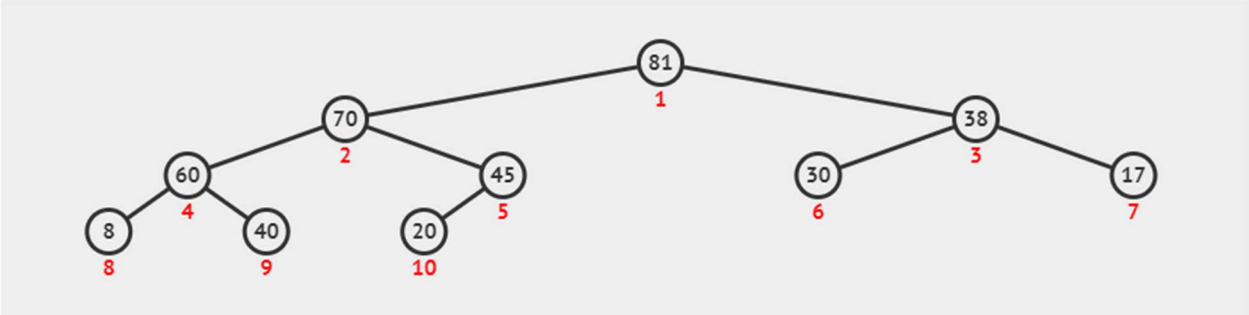
Q.7 [C311.1] [1M] Which of the following is/are not a correct inorder traversal sequence(s) of binary search tree(s)?

I. 2, 4, 6, 7, 14, 18, 24
II. 4, 7, 8, 11, 9, 16, 24
III. 1, 6, 9, 7, 13, 15, 19
IV. 3, 5, 6, 8, 17, 19, 24

Answer: II and III

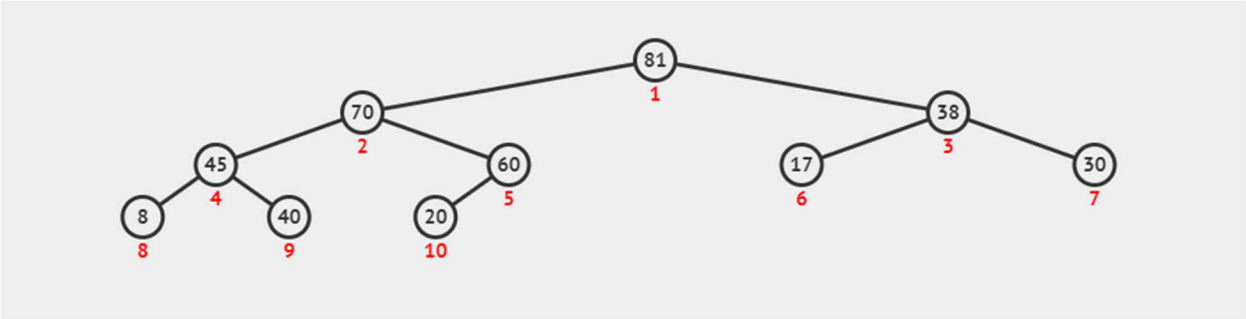
Q.8 [C311.1][1M] Insert the following values in max heap and show the final tree
8, 20, 17, 40, 45, 30, 38, 60, 70, 81

Answer:



Tree insertion one by one

OR



Tree construction and heapify

Q.9 [C311.1][2M] If n is the number of nodes in a binary tree and i is initialized as 0. Assuming that every key in a binary tree is unique then fill in the blanks when the above function checks for whether a given binary tree is max heap tree or not. Do not make any new function. Fill only the blanks value.

```
bool  check(struct node  *root, int i, int n)
{
   if (root == NULL) {
      return true;
   }
  …….. (i)
   if ((root->left && root->left->data > root->data) ||
         (root->right && root->right->data > root->data)) {
      return false;
   }
   ……(ii)
}
```

Answer:

First blank:

```
if (i >= n) { return false; }
```

Second blank

```
return check(root->left, 2*i + 1, n) && check(root->right, 2*i + 2, n);
```

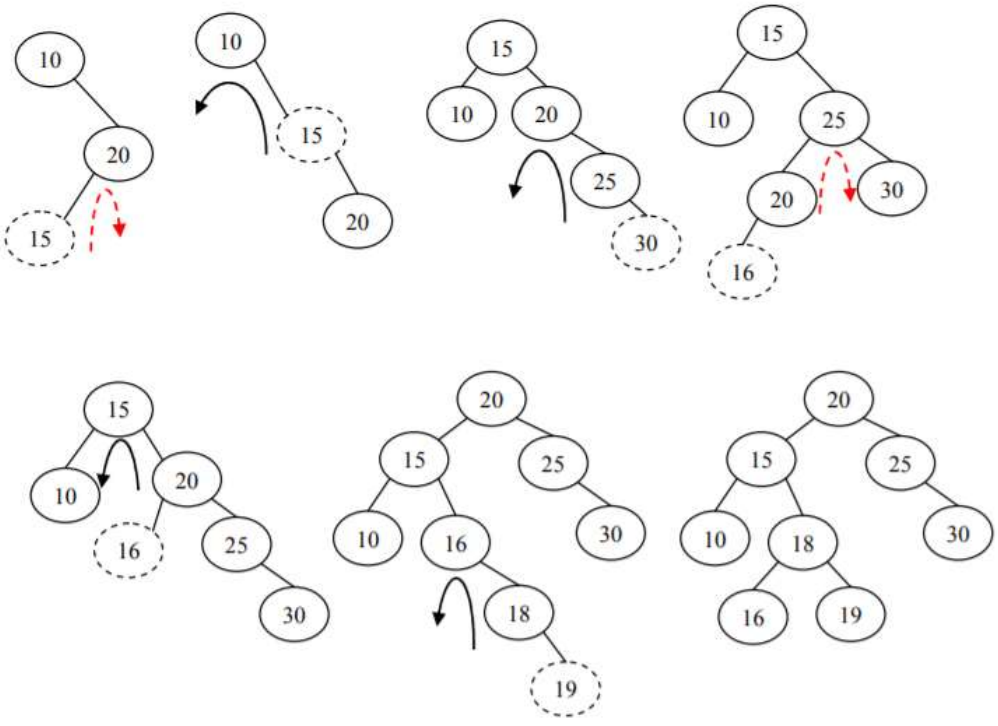
Q.10 [C311.1]

a) [1M] Insert the following sequence of elements into an AVL tree, starting with an empty tree: 10, 20, 15, 25, 30, 16, 18, 19.

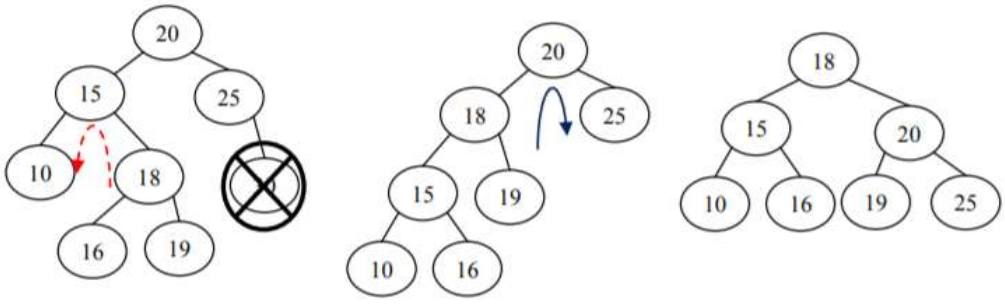
b) [1M] Delete 30 in the AVL tree that is obtained in part (a).

Answer:

(a) Red dashed line signifies first part of double rotate action.



(b).

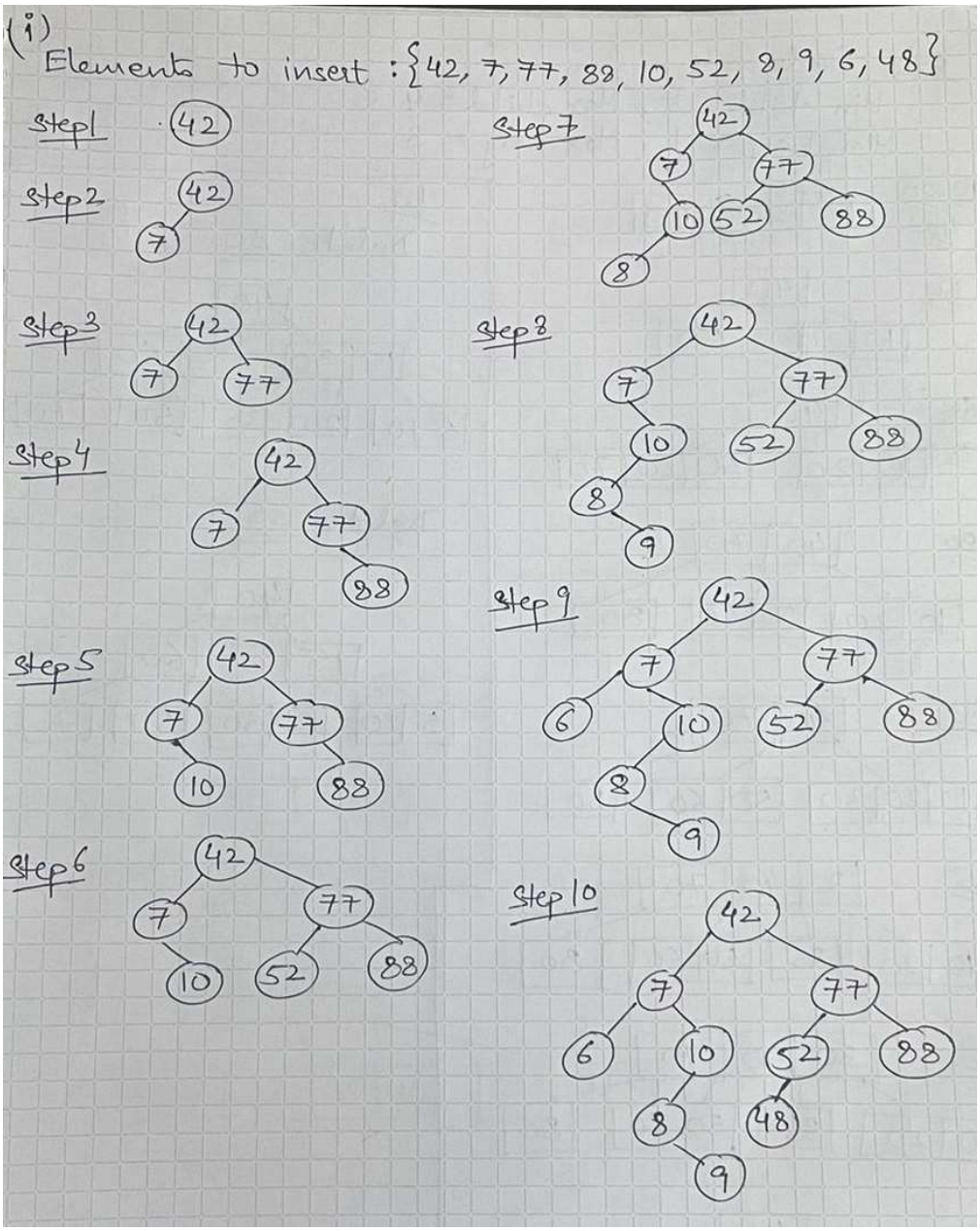


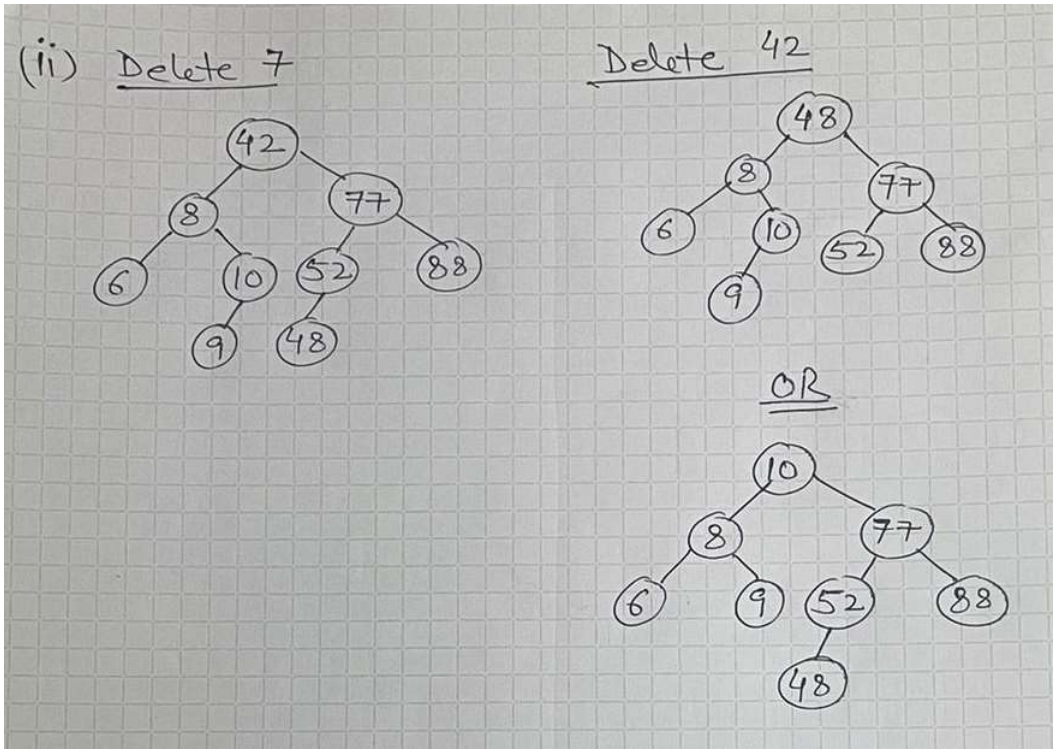
Q.11 [C311.1]

a) [1M] Draw the BST (Binary Search Tree) that results from inserting the values given below (reading from left to right): 42, 7, 77, 88, 10, 52, 8, 9, 6, 48 (Show all insertion steps one by one).

b) [1M] Delete 7 and afterwards delete 42. If possible, make sure that the number of nodes in the left and right subtree of the root node are equal. Show BST after each deletion.

Answer:





Q.12 [C311.1] Consider the following elements needs to form a B tree of order 4.
10, 20, 40, 50, 60, 70, 80, 30, 35, 5, 15

a) [2M] insert the elements in given order to create a B tree.
b) [1M] Delete the elements 80 followed by 35 from the above constructed tree.

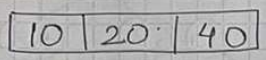
Answer:

a) Order = 4

Min child = 2, Max child = 4

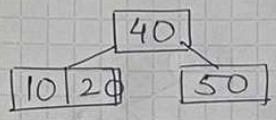
Min key = 1, Max key = 3

10 20 40



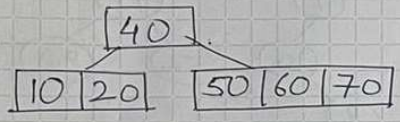
40

50

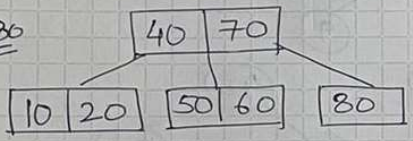


60

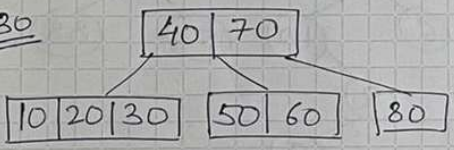
70



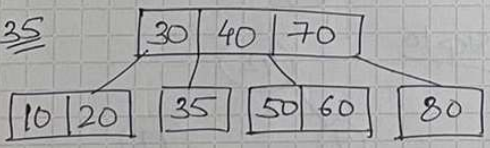
80



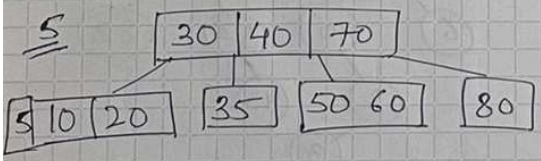
30



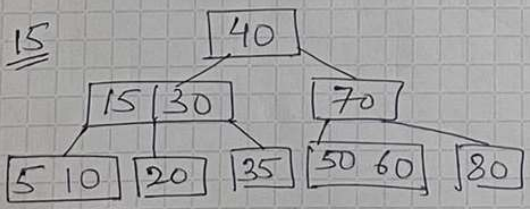
35



5

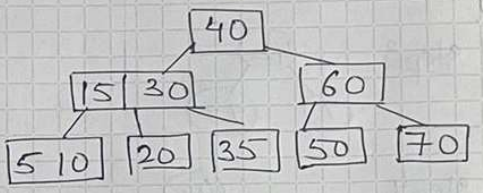


15

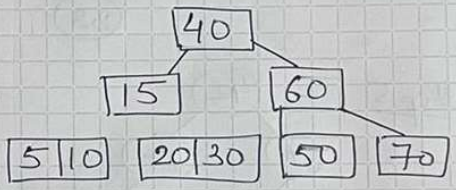


b)

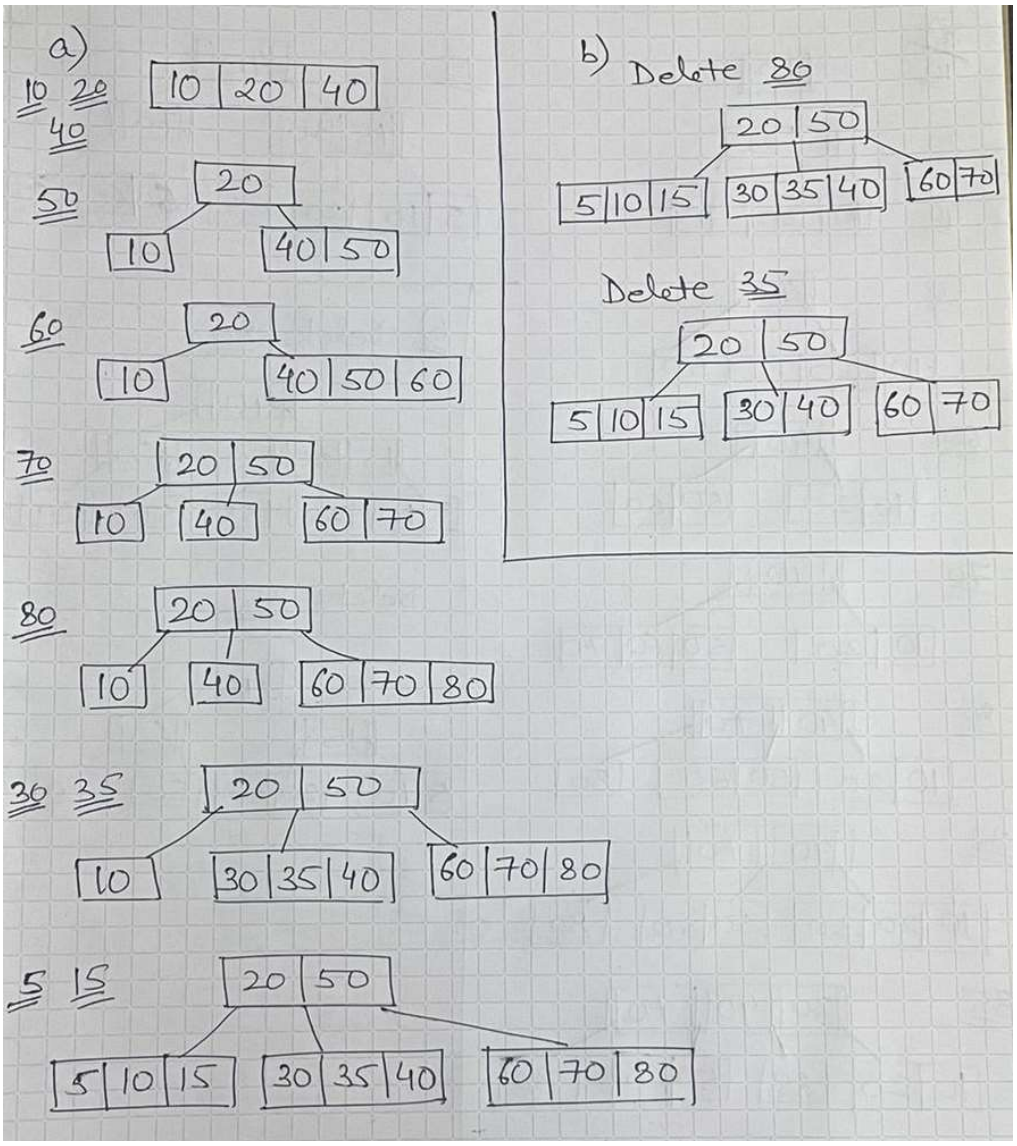
Delete 80



Delete 35



OR



Q.13 [C311.3] Consider the following adjacency matrix:

	a	b	c	d	e	f	g
a	0	1	0	1	0	0	0
b	1	0	1	1	1	0	0
c	0	1	0	0	1	1	0
d	1	1	0	0	1	0	1
e	0	1	1	1	0	1	1
f	0	0	1	0	1	0	1
g	0	0	0	1	1	1	0

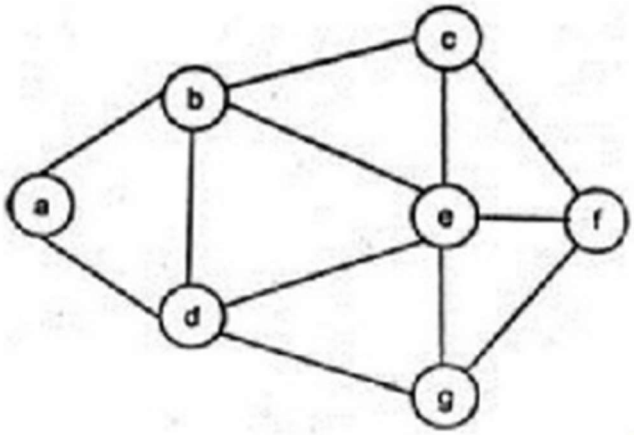
I. [1M] Draw the labeled graph for given adjacency matrix.

II. [1M] Show step by step BFS (Breadth First Search) traversal of obtained graph using suitable data structure. Use vertex “a” as starting vertex.

III. [1M] Show step by step DFS (Depth First Search) traversal obtained graph using suitable data structure. Use vertex "a" as starting vertex.

Answer:

I.



Multiple answer possible for BFS and DFS. One of the answer is-

II. *Using queue data structure*

Enqueue: a

a						

Dequeue: **a**, Enqueue: b, d

	b	d				

Dequeue: **b**, Enqueue: c, e

		d	c	e		

Dequeue: **d**, Enqueue: g

			c	e	g	

Dequeue: **c**, Enqueue: f

				e	g	f

Dequeue: **e**

					g	f

Dequeue: **g**

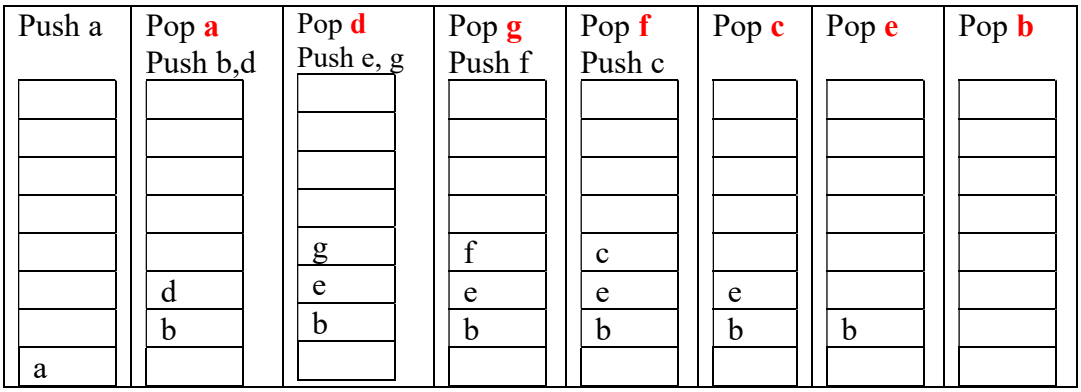
						f

Dequeue: **f**

BFS: a,b,d,c,e,g,f

III.

Using stack data structure



DFS: a,d,g,f,c,e,b