50

20 60

10 40 70

15 30 65 80

25 35 75

b)

50

10 60

40 70

35 65 80

25 75

15

Pre order Traversal:

50 20 10 15 40 30 25 35 60 70 65 80 75

In order Traversal :

10 15 20 25 30 35 40 50 60 65 70 75 80

Post Order Traversal:

15 10 25 35 30 40 20 65 75 80 70 60 50

1. a)

7

3 5

1 2 4

b)

arr[0] 7

arr[1] 3

arr[2] 5

arr[3] 1

arr[4] 2

arr[5] 4

c)

arr[0] 5

arr[1] 3

arr[2] 4

arr[3] 1

arr[4] 2

3 a)

struct Node

{

Node \* leftChild;

Node \* rightChild;

Node \* parent;

int val;

}

b)

insert(Node \* p, Node\* root)

{

if root is nullptr

set p to root and p’s parent to nullptr

else if p->val is less than root->val

if root doesn’t have a leftChild

set root’s leftChild to p and p’s parent to root

else insert(p,root->leftChild)

else if p->val is greater than root->val

if root doesn’t have a rightChild

set root’s rightChild to p to and set p’s parent to root

else insert(p,root->rightChild)

}

4

1. O (S+C)
2. O(logS +C)
3. O(log C + log S)
4. O(logS)
5. O(1)
6. O(log C +S)
7. O(SlogS)
8. O(ClogS)