Homework 5

1 a]

50

20 60

10 40 70

15 30 64 80

23 36 75

1 b] Preorder : 50,20, 10, 15, 40, 30, 23, 36, 60, 70, 64, 80, 75

Inorder : 10, 15, 20, 23, 30, 36, 40, 50, 60, 64, 70, 75, 80

PostOrder: 15, 10, 23, 36, 30, 40, 20, 64, 75, 80, 70, 60, 50

1 c]

50

20 60

10 40 70

15 23 64 80

36 75

After step 1

50

15 60

10 40 70

23 64 80

36 75

After step 2

2 a]

struct Node{

int vals;

Node\* left;

Node\* right;

Node\* parent;

}

2 b] void insertIntoTree(int x)

if tree is empty

create a new node and make the root point to it. Left, right and parent node are still nullptr

return;

starting from root, visit every integer in the tree

if value of a node equals x, do nothing and return;

if value of a node is more than x,

check if there is a left node, if yes access the left node and continue checking recursively. If no left node present, create a new node on the node’s left child. return;

if value of node is less than x,

check if there is a right node, if yes access the right node and continue checking recursively. If no right node present, create a new node on the node’s right child. return;

3 a]

7

5 6

3 1 2

b]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 5 | 6 | 3 | 1 | 2 |

c]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 5 | 2 | 3 | 1 |

4 a] O(C + log S)

b] O(log C + S)

c] O(log C + log S)

d] O(log S) [technically log S + 1]

e] O(1)

f] O(log C + S)

g] O(S \* log S )

h] O(C \* log S)