1a:

50

20 60

10 40 70

15 30 63 80

24 36 75

1b:

Inorder: 10 15 20 24 30 36 40 50 60 63 70 75 80

Preorder: 50 20 10 15 40 30 24 36 60 70 63 80 75

Postorder: 15 10 24 36 30 40 20 63 75 80 70 60 50

1c:

50

15 60

10 40 70

24 63 80

36 75

2a:

struct Node

{

int value;

Node\* left, right, parent = nullptr;

};

2b:

void insert (int v, Node\* root)

{

if (root == nullptr)

{ allocate new node with value v;

have root point to this node, return; }

Node\* ptr = root;

if (v == ptr->value) return; //value already exists

if (v < ptr->value)

{

if (ptr->left == nullptr)

{

allocate new node N with value v;

ptr->left = N;

N-> parent = ptr; //current node becomes new node’s parent

return;

}

insert(v, ptr->left); //if not nullptr, continue traversing tree, recursive

}

if (v > ptr->value)

{

if (ptr->right == nullptr)

{

allocate new node N with value v;

ptr->right = N;

N-> parent = ptr; //current node becomes new node’s parent

return;

}

insert(v, ptr->right); //if not nullptr, continue traversing tree, recursive

}

}

3a:

7

5 6

3 0 4

3b:

7 5 6 3 0 4

3c:

6 5 4 3 0

4a: O(c + s)

4b: O(log c + s)

4c: O(log c + log s)

4d: O(log s)

4e: O(1)

4f: O(log c + s)

4g: O(s)

4h: O(c log s)