1a.

50

20

10

xx

15

40

30

25

36

xx

60

xx

70

65

80

74

xx

1b.

In-Order: 10 15 20 25 30 36 40 50 60 65 70 74 80

Pre-Order: 50 20 10 15 40 30 25 36 60 70 65 80 74

Post-Order: 15 10 25 36 30 40 20 65 74 80 70 60 50

1c.

50

15

10

40

25

xx

36

xx

60

xx

70

65

80

74

xx

2a.

struct BinaryTreeNode

{

int val;

Node\* leftchild;

Node\* rightchild;

Node\* parent;

};

2b.

input (Node with value val)

If the tree is empty

Allocate a new node with target value

Set the root pointer to new node

Start traversing from the root of the tree

While the new node has not been inserted

if val is equal to the current node’s value

return (do nothing)

if val is less than the current node’s value

set parent node to current node

if current node has a leftchild node

set current node to the leftchild node (keep advancing towards left)

else

allocate a new node with value val

set current node to new node

return

if val is greater than the current node’s value

set parent node to current node

if current node has a rightchild node

set current node to rightchild node (keep advancing towards right)

else

allocate a new node with value val

set current node to new node

3a.

8

3

0

2

6

4

3b. 836024

3c. 63402

4a. O(C + S)

4b. O(logC + S)

4c. O(logC + logS)

4d. O(logS)

4e. O(1)

4f. O(logC + S)

4g. O(SlogS)

4h. O(ClogS+C)