1a.

50

20 60

10 40 70

65 80

30 45 75

25 35

1b.

50

25 60

10 40 70

35 45 65 80

75

1c. post-order = 10, 25, 35, 30, 45, 40, 20, 65, 75, 80, 70, 60, 50

in-order = 10, 20, 25, 30, 35, 40, 45, 50, 60, 65, 70, 75, 80

pre- order = 50, 20, 10, 40, 30, 25, 35, 45, 60, 70, 65, 80, 75

go back

2a.

6

3 5

1 2 4

2b. 6, 3, 5, 1, 2, 4

2c. 5, 3, 4, 1, 2

3a.

struct Node

{

Node\* parent;

Node\* leftChild;

Node\* rightChild;

int value;

}

3b.

insert(Node\* &tree, int value, Node\* parent)

{

if empty

set root pointer equal to the newly allocated node, made with value

return

loop through the nodes starting from the root

while loop has not been terminated by return

if the current node’s value is the same as the inputted value

return

if the inputted value is smaller than the value of the current node

if current node’s left child is null

set current node’s left child to newly allocated node pointer with given value

set new node’s parent pointer to current node

return

else

continue in the left direction

if the inputted value is greater than the value of the current node

if the current node’s right child is null

set current node’s right child to newly allocated node pointer with given value

set new node’s parent pointer to current node

return

else

continue in right direction

}