1a.

50

20 60

10 40 70

30 45 65 80

25 35 75

1b.

50

10 60

40 70

25 45 65 80

35 75

1c. In-order: 10 20 25 30 35 40 45 50 60 65 75 80

Pre-order: 50 20 10 40 30 25 35 45 60 70 65 80 75

Post-order: 10 25 35 30 45 40 20 65 75 80 70 60 50

2a.

6

3 5

1 2 4

2b. array, from 0 to count:

6 3 5 1 2 4

2c. array, from 0 to count:

5 3 4 1 2

3.

struct Node

{

int value;

Node\* parent;

Node\* left;

Node\* right;

};

For some function prototype void insert(Node\* node, int value):

1. If node’s parent pointer is not nullptr, examine the parent node instead of the given node and go back to step 1
2. If value is equal to node->value, return
3. If value is less than node->value, examine the left node pointer
4. If value is greater than node->value, examine the right node pointer
5. If the examined node pointer is nullptr, allocate a new Node, fill in its values, set the node pointer to this node, and then return
6. If the examined node pointer is not nullptr, go back to step 2 using this node