1. a.

50

20

10

40

30

25

35

45

60

xx

70

65

80

75

xx

b. 50

10

xx

40

25

xx

35

45

60

xx

70

65

80

75

xx

c. 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

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

2. a.

6

3

1

2

5

4

x

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

c. 6,3,4,1,2

3. a.

struct Node

{

Node(cons tint value)

{

m\_value=value;

m\_left=nullptr;

m\_right=nullptr;

m\_parent=nullptr;

}

int m\_value;

Node\* left;

Node\* right;

Node\* parent;

};

b. pseudocode:

if tree is empty

create a new node

let root pointer points to it

return

otherwise

create a new node pointer

let it points to the root node

repeated infinitely

if the current node already contains the value

return

if current node has a value smaller than the inserted value

if current node has a left child

pointer points to its left child

otherwise

create a new node, use the inserted value for the argument for the

constructor

make it the left child of the node that the pointer currently points to

set its parent pointer to the current node

return

otherwise

if current node has a value larger than the inserted value

if current node has a right child

pointer points to its right child

otherwise

create a new node, use the inserted value for the argument for the

constructor

make it the right child of the node that the pointer currently points to

set its parent pointer to the current node

return