1.

a. 50

20 60

10 40 70

30 45 65 80

25 35 75

b. 50

10 60

40 70

25 45 65 80

35 75

c.

In-order traversal:

10 20 25 30 35 40 45 50 60 65 70 75 80

Pre-order traversal:

50 20 10 40 30 25 35 45 60 70 65 80 75

Post-order traversal:

10 25 35 30 45 40 20 65 75 80 70 60 50

2.

a. 6

3 5

1 2 4

b. The heap in part a can be represented as an array: 6 3 5 1 2 4

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

3.

a. struct TreeNode

{

int m\_data;

TreeNode\* leftChildPtr;

TreeNode\* rightChildPtr;

TreeNode\* parentPtr;

}

b.

void insertItem (Node\* parentPtr, int newData)

if the parentPtr equals to nullptr:

Create a new node.

Copy newData into new node’s m\_dara.

Set all the pointers in the new node to nullptr.

Make the parentPtr point to this new node.

else if the leftChildPtr of parentPtr is a nullptr and the rightChildPtr of

parentPtr is also a nullptr

Create a new node.

Copy newData into new node’s m\_data.

Set the pointer parent of the new node to parentPtr.

Set the pointers leftChild and rightChild of the new node to nullptr.

If newData is bigger than the m\_data of node pointed by parentPtr,

make the rightChildPtr of the parent node point to the new node

If newData is not bigger than the m\_data of node pointed by parentPtr,

make the leftChildPtr of the parent node point to the new node

else if only the rightChildPtr of parentPtr is a nullptr

if newData is bigger than the m\_data of node pointed by parentPtr

make the rightChildPtr of the parent node point to the new node

else execute the recursion function insertItem(parentPtr->leftChildPtr,

newData)

else if only the leftChildPtr of parentPtr is a nullptr

if newData is not bigger than the m\_data of node pointed by parentPtr

make the leftChildPtr of the parent node point to the new node

else execute the recursion function insertItem(parentPtr->rightChildPtr,

newData)

else if both children pointers of parentPtr are not nullptrs

if newData is bigger than the m\_data of node pointed by parentPtr

execute the recursion function insertItem(parent->rightChildPtr,

newData)

else execute the recursion function insertItem(parentPtr->leftChildPtr,

newData)