1a.

1b.

in-order: 10, 15, 20, 25, 30, 35, 40, 50, 60, 65, 70, 75, 80

pre-order: 50, 20, 10, 15, 40, 30, 25, 35, 60, 70, 65, 80, 75

post-order: 15, 10, 25, 35, 30, 40, 20, 65, 75, 80, 70, 60, 50

1c.

2a.

struct Node

{

int data;

Node\* leftChild;

Node\* rightChild;

Node\* parent;

};

2b.

declare a new Node assign with the value we want to insert

if head is null

return

while current node is not nullptr

if value of new node less than value of current node

if leftChild of current node is nullptr

set new Node's parent pointer point to current Node

assign new Node to leftChild of current node

return

else

set current node to left child of current node

if value of new node greater than value of current node

if rightChile of current node is nullptr

set new Node's parent pointer point to current Node

assign new Node to rightChild of current node

return

else

set current node to right child of current node

if value of new node equal to value of current node

return

3.a.

3.b.

array a = {8, 3, 6, 0, 2, 4}

3.c.

array b = {6, 3, 4, 0, 2}

4.

a. O(C + logS)

b. O(logC + S)

c. O(logC + logS)

d. O(C + logS)

e. O(C + S)

f. O(logC + S)

g. O(C + SlogS)

h. O (C \* logS)