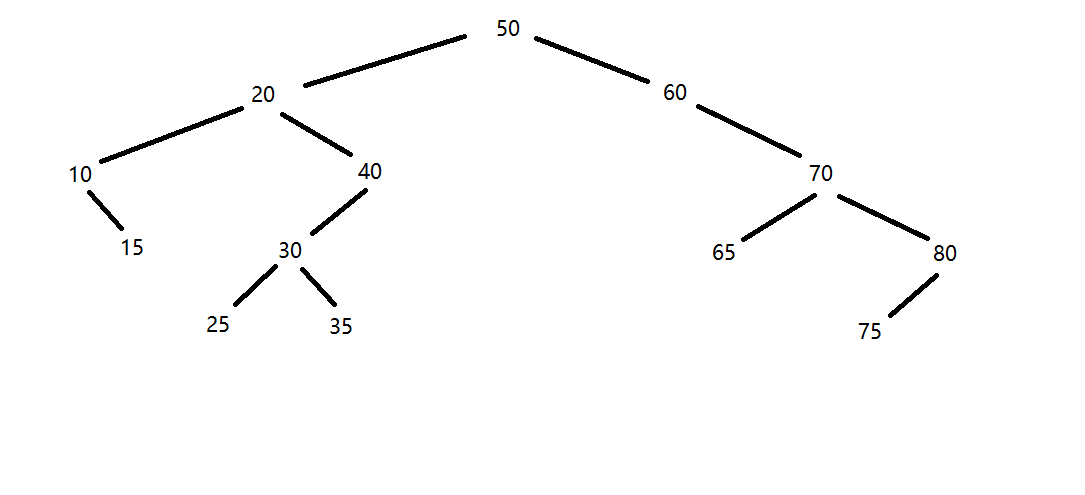
1a).



1b).

Pre-order:

50 20 10 15 40 30 25 35 60 70 65 80 75

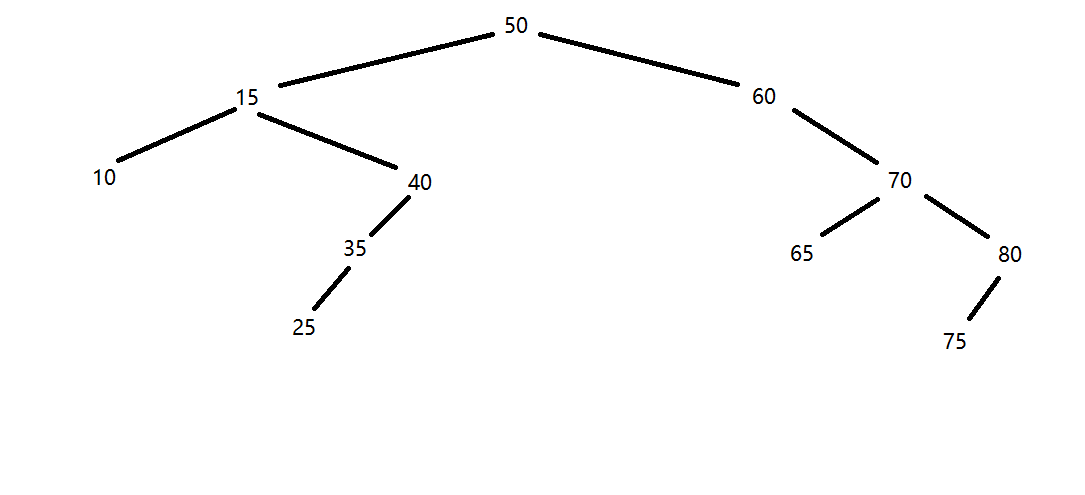
In-order:

10 15 20 25 30 35 40 50 60 65 70 75 80

Post-order:

15 10 25 35 30 40 20 65 75 80 70 60 50

1c).



2a).

struct binarySearchTreeNode

{

int value;

binarySearchTreeNode\* parent;

binarySearchTreeNode\* left;

binarySearchTreeNode\* right;

};

2b).

insertItem (binarySearchTreeNode\* curr, int value, binarySearchTreeNode\* parent)

if curr is a null pointer

create a new binary search tree node, assign value to the new node, and assign parent to the parent pointer of the new node.

let the left and right pointers of the new node be nullptr.

let curr points to the new tree node.

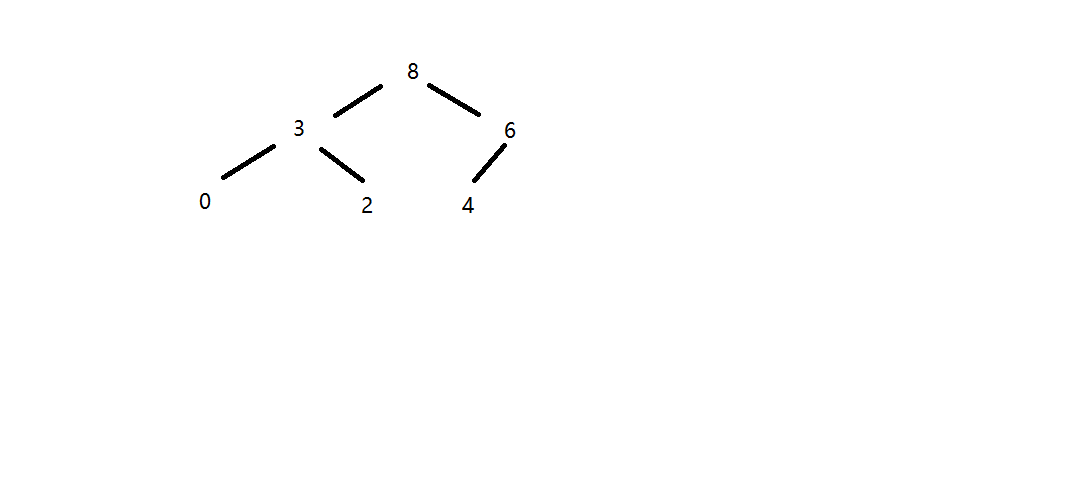
else if value is less than curr’s value

insertItem(curr’s left, value, curr)

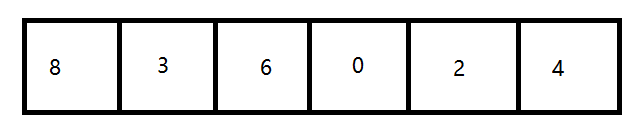
else if value is greater than curr’s value

insertItem(curr’s right, value, curr)

3a).

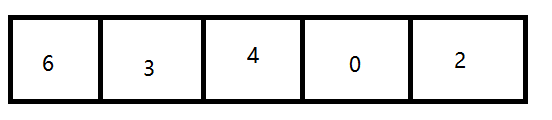


3b).



Count: 6

3c).



Count: 5

4).

a. O(S + C)

b. O(log(C) + S)

c. O(log(C) + log(S))

d. O(log(S))

e. O(1)

f. O(S + log(C))

g. O(S)

h. O(C \* log(S))