1.

a.

50

20 60

10 40 70

15 30 65 80

25 35 75

b.

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

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

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

c.

delete node 30

50

20 60

10 40 70

15 25 65 80

35 75

Delete node 20

50

15 60

10 40 70

25 65 80

35 75

2.

a.

struct Node

{

Node\* left, right, parent;

Int m\_value;

}

b.

void insert(int value)

{ insertHelper(BSTRoot, value, nullptr); }

Void insertHelper(Node\* curr, int value, Node\* parent)

{

if (curr is nullptr)

create a new Node object with the data as value, parent pointer as parent, and left and right pointer as nullptr, then return;

else if (curr’s value is smaller than value)

insertHelper(curr’s left child, value, curr);

else insertHelper(curr’s right child, value, curr);

}

3.

a.

8

3 6

0 2 4

b.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 8 | 3 | 6 | 0 | 2 | 4 |

Count = 6

c.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 3 | 4 | 0 | 2 |

Count = 5

4.

a. O(c+s)

b. O(logc + s)

c. O(logc + logs)

d. O(logs)

e. O(1)

f. O(s+logc)

g. O(s)

h. O(clogs)