1a)

50

25

30

35

15

75

65

80

70

60

20

40

10

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) After deleting 30 and 20

50

75

65

80

70

60

15

40

10

25

35

2

a)

struct Node

{

Node(int value, Tree \*parent) : m\_parent(parent), m\_value(value), left(nullptr), right(nullptr)

{

}

int m\_value;

Node \*left;

Node \*right;

Node \*m\_parent;

};

b)

void insert(Tree \*add, int value, Tree \*parent)

{  
 if(add is a nullptr)

Create a new Tree node and add here. The parent pointer is the this parent pointer and the left pointer and the right pointer are nullptr.

return

Else if(value is greater than the current value)

Insert(add->right, value, parent)

Else if(value is less than the current value)

insert(add->left, value, parent)

}

3

4

8

6

2

0

3

a)

b)

8 3 6 0 2 4

c)

6 3 4 0 2

4

a)

O(C+S)

b)

O(logC+S)

c)

O(logC+logS)

d)

O(S)

e)

O(1)

f)

O(logC+S)

g)

O(S)

h)

O(ClogS)