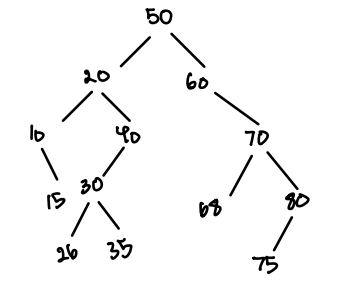
1a.



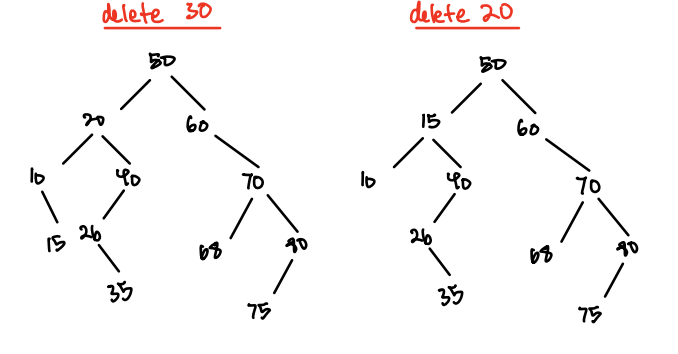
1b.

preorder: 50, 20, 10, 15, 40, 30, 26, 35, 60, 70, 68, 80, 75

postorder: 10, 15, 26, 35, 30, 40, 20, 68, 75, 80, 70, 60, 50

inorder: 10, 15, 20, 26, 30, 35, 40, 60, 68, 70, 75, 80

1c.



2a. struct BST{

int val;

BST\* left;

BST\* right;

BST\* parent;

BST(inv v, BST\* p): val(v), left(nullptr), right(nullptr), parent(p){}

}

2b.

void Insert(BST\* root, int value){

if root == nullptr

Create New Node and set value

Point root to New node

Set left pointer to nullptr

Set right pointer to nullptr

Set parent to null

while (root != nullptr){

if (value < root->data){

if (root->left != nullptr)

go to left node

else{

Add New Node and set value

Point root to New node

Set left pointer to nullptr

Set right pointer to nullptr

Set parent to root

}

}

if (value < root->data){

if (root->right != nullptr)

go to right node

else{

Add New Node and set value

Point root to New node

Set left pointer to nullptr

Set right pointer to nullptr

Set parent to root

}

}

else{

//Values are equal

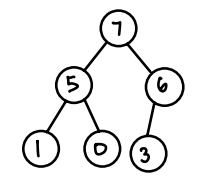
return (Do Nothing);

}

}

}

3a.



3b.



3c.



4

a. O(C + S)

b. O(log(C) + S)

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

d. O(log(S))

e. O(1)

f. O(log(C) + S)

g. O(S\*log(S))

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