Joshua Liu

105136031

CS 32

Homework 5

A close up of a map

Description automatically generated

1b.

in-order: 10 15 20 25 30 39 40 50 60 65 70 71 80

pre-order: 50 20 10 15 40 30 25 39 60 70 65 80 71

post-order: 15 10 25 39 30 40 20 65 71 80 70 60 50

A close up of a map

Description automatically generated

2a.

struct TreeNode

{

int data;

TreeNode\* parent;

TreeNode\* leftChild;

TreeNode\* rightChild;

};

2b.

Inserting new node in binary search tree with parent pointers

Input item to insert

if tree is empty

create new node with item as data

point root node to this new node

set parent pointer to null pointer

begin at root of tree, while not done inserting

if item is equal to current node’s item

return

if item is less than current node’s item

if there is a left child

go to that child and run again

else

create new node with item as data and set current node’s left child to new node

set new node’s parent pointer to current node

if item is greater than current node’s item

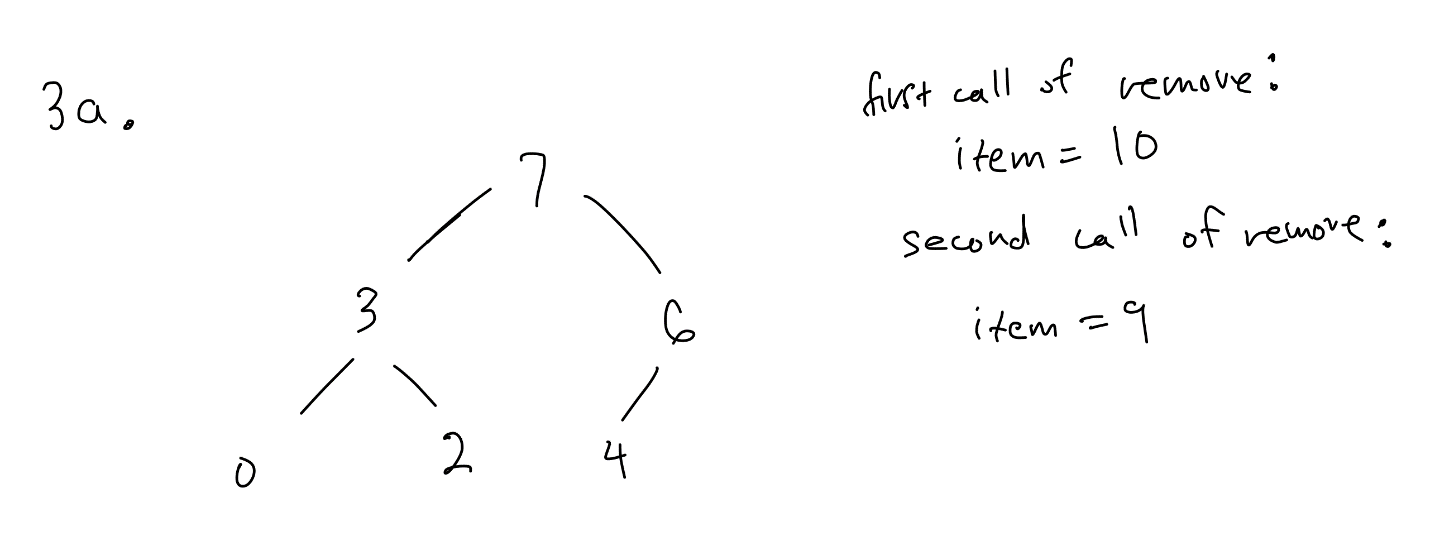
if there is a right child

go to that child and run again

else

create new node with item as data and set current node’s left child to new node

set new node’s parent pointer to current node

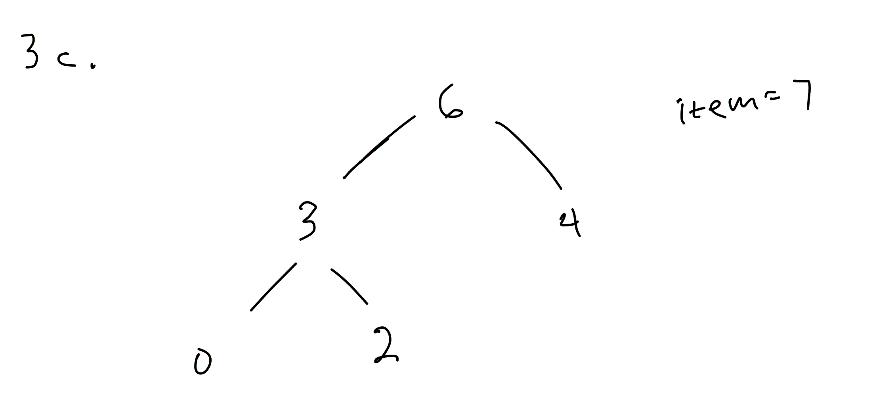


3b.

Position 0 1 2 3 4 5

Value 7 3 6 0 2 4

* 7 is top item
* 3 and 6 are 7’s children
* 3’s children are at position (1\*2 + 1) and (1\*2 + 2), which are positions 3 and 4 ( which are of value 0 and 2)
* 6’s child is at position (2\*2 + 1) which is 5 (value 4). It only has one child since (2\*2 + 2) = 6, and there are no more values past position 5.



4a. O(S + C)

4b. O(log C + S)

4c. O(log C + log C)

4d. O(1 + log S)

4e. O(1 + 1)

4f. O(log C + log S)

4g. O(1 + S log S)

4h. O(C log S)