Madeleine Murphy

004991129

CS 32 Homework 5

Due June 6, 2019

1a.

50

20 60

10 40 70

15 30 65 80

25 39 71

1b. Pre-order traversal: 50 20 10 15 40 30 25 39 60 70 65 80 71

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

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

1c.

50

15 60

10 40 70

25 65 80

39 71

Delete 30, can be replaced by either 25 (largest in left) or 39 (smallest in right). Delete 20, can be replaced by either 15 (largest in left) or 25 (smallest in right)

2.

a.

struct TreeNode

{

int data;

TreeNode \*left, \*right, \*parent;

};

b.

insert a Node

If the tree is empty

Point root pointer to our new node

Left, right, parent = Nullptr

Start at the root of the tree

While we still have not added value

If data in our node = current Node

Set new node parent to current Node

If there’s a left child

left child’s parent = new node

new node’s left = left child

current node’s left = new node

new node right = null

If data in new node is less than current Node’s value

If there is a left child, go left

Else

Current Node left = new node

new Node’s parent = current node

Return

If V is greater than current Node’s value

If there is a right child, go right

Else

Current node right = new node

New Node’s parent = current node

Return

3. h.insert(3);

h.insert(6);

h.insert(2);

h.insert(0);

h.insert(10);

h.insert(4);

int item;

h.remove(item); // Removes the biggest item from the heap, and puts it in item

h.insert(9);

h.insert(7);

h.remove(item);

a.

7

3 6

0 2 4

b.

7

3

6

0

2

4

c.

6

3

4

0

2

4.

a. O(C + S)

b. O(log2C + S)

c. (log2C + log2S)

d. O(log2S)

e. O(1)

f. O(log2C + S)

g. O(SlogS)

h. O(Clog2S)