Com Sci 32

Homework 5

# Problem 1

Problem 1a

Problem 1b

In-order traversal –

10 15 20 25 30 39 40 50 60 65 70 71 80

Pre-order traversal –

50 20 10 15 40 30 25 39 60 70 65 80 71

Post-order traversal –

15 10 25 39 30 40 20 65 71 80 70 60 50

Problem 1c

Although several alternatives are available, the following route was chosen, displayed step by step:

After removal of 30 –

After removal of 20 -

# Problem 2

Problem 2a

struct Node

{

int data;

Node\* left;

Node\* right;

Node\* parent;

}

Problem 2b

void insert(Node\* curr, Node\* toAdd)

if toAdd’s data is less than curr’s data

if curr’s left pointer is a null pointer

set curr’s left pointer to toAdd and set toAdd’s parent pointer to curr

otherwise, recursively call insert for curr’s left pointer and toAdd

if toAdd’s data is greater than curr’s data

if curr’s right pointer is a null pointer

set curr’s right pointer to toAdd and set toAdd’s parent pointer to curr

otherwise, recursively call insert for curr’s right pointer and toAdd

Note - we expect the user’s call to this function to have arguments (root, toAdd), where root is a pointer to the root node of the BST, and toAdd is a pointer to a well-formed instance of the Node struct (i.e. it should have its left and right pointers set to nullptr, and should have a valid value for its data member variable).

# Problem 3

Problem 3a

Problem 3b

In array form, the heap would be - [7, 3, 6, 0, 2, 4]

Problem 3c

After removing one more item from the heap, it would look like - [6, 3, 4, 0, 2]

# Problem 4

4a – O(C + S)

4b – O(log(C) + S)

4c – O(log(C) + log(S)) = O(log(CS))

4d – O(log(S))

4e – O(1)

4f – O(log(C) + S)

4g – O(Slog(S))

4h – O(Clog(S))