Raj Kumar

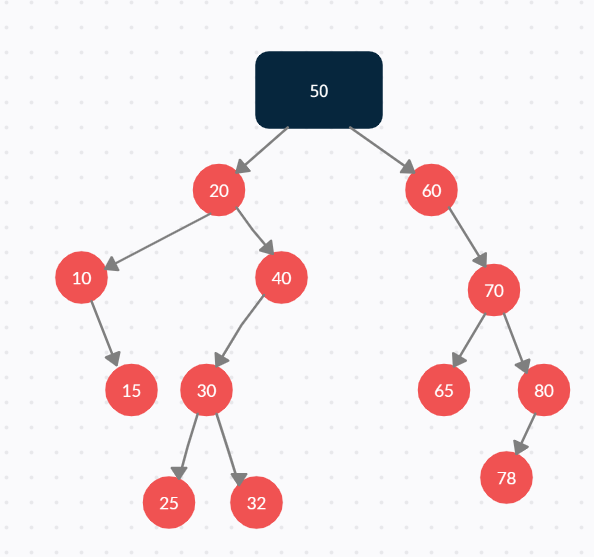
Professor Smallberg

Computer Science 32

11 March 2021

Homework 4

**Question 1a:**



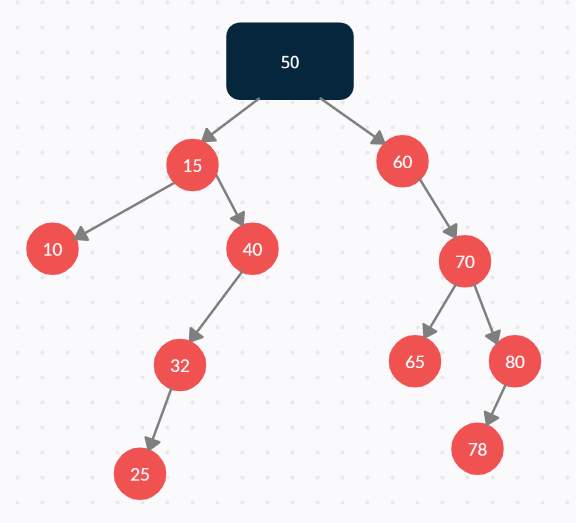
**Question 1b: After inserting the nodes mentioned in part a, what would be printed out by inorder, preorder, and postorder traversals of the tree (assume your traversal function prints out the number at each node as it is visited)?**

Preorder: 50, 20, 10, 15, 40, 30, 25, 32, 60, 70, 65, 80, 78

Inorder: 10, 15, 20, 25, 30, 32, 40, 50, 60, 65, 70, 78, 80

Postorder: 15, 10, 25, 32, 30, 40, 20, 65, 78, 80, 70, 60, 50

**Question 1c: After inserting the nodes mentioned in part a, what is the resulting BST after you delete the node 30, then the node 20? (Again, just use a simple deletion algorithm with no balancing. If you have an option of making a choice, any correct choice is acceptable.)**



**Question 2: In some binary search trees, each node has a left child pointer, a right child pointer and a parent pointer. The parent pointer of a node points to its parent (duh!), or is nullptr if the node is the root node. This problem will examine such trees.**

**Question 2a: Show a C++ structure/class definition for a binary tree node that has both child node pointers and a parent node pointer. Assume the data stored in each node is an int.**

struct BSTNode

{

BSTNode\* parent;

BSTNode\* left;

BSTNode\* right;

int data;

};

**Question 2b: Write pseudocode to insert a new node into a binary search tree with parent pointers. (Hint: You can find binary search tree insertion code on pp. 471-473).**

Make new node with appropriate int data initialized

Store nullptr into parent variable in new node

Make pointer called current node that points to the root of the BST

Repeatedly (loop)

If (current node’s pointer is nullptr)

Check what side (left or right) you followed on parent node, and set the parent’s corresponding left or right pointer to pointer to new node

Exit loop

Store current node’s pointer into parent variable in new node

If (new node’s data is less than current node’s data)

Set current node to current node’s left pointer

Else if(new node’s data is the same as current node’s data)

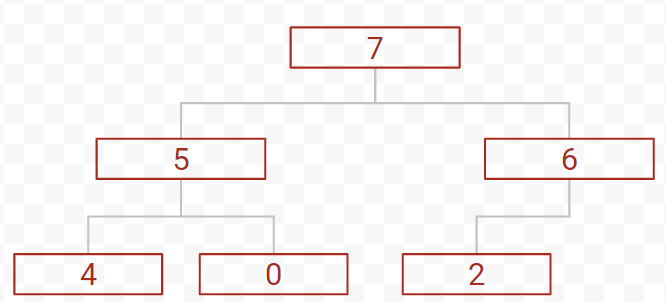
Delete new node

Exit loop

else

Set current node to current node’s right pointer

**Question 3a: Show the resulting heap (As in problem 1a, show the tree in some recognizable form.)**



**Question 3b: Show how your heap from part a would be represented in an array.**

Array: 7, 5, 6, 4, 0, 2

**Question 3c: After executing h.remove(item); one more time, show the array that results.**

Array: 6, 5, 2, 4, 0

**Question 4a: vector<pair<string, list<int>>>, where each pair in the outer vector represents a course and all the students in that class, with those students being sorted in order. The pairs are in no particular order in the outer vector. What is the big-O complexity to determine whether a particular student s is enrolled in course c?**

**4a: O(C + S)**

**4b: O(log C + S)**

**4c: O(log C + log S)**

**4d: O(1 + log S) or O(log S)**

**4e: O(1 + 1) or O(1)**

**4f: O(log C + S)**

**4g: O(1 + S log (S)) or O(S log (S))**

**4h: O(C \* log S)**