

Sümeyye ACAR

ID: 22103640

CS202 / 001

HW2

Question 1

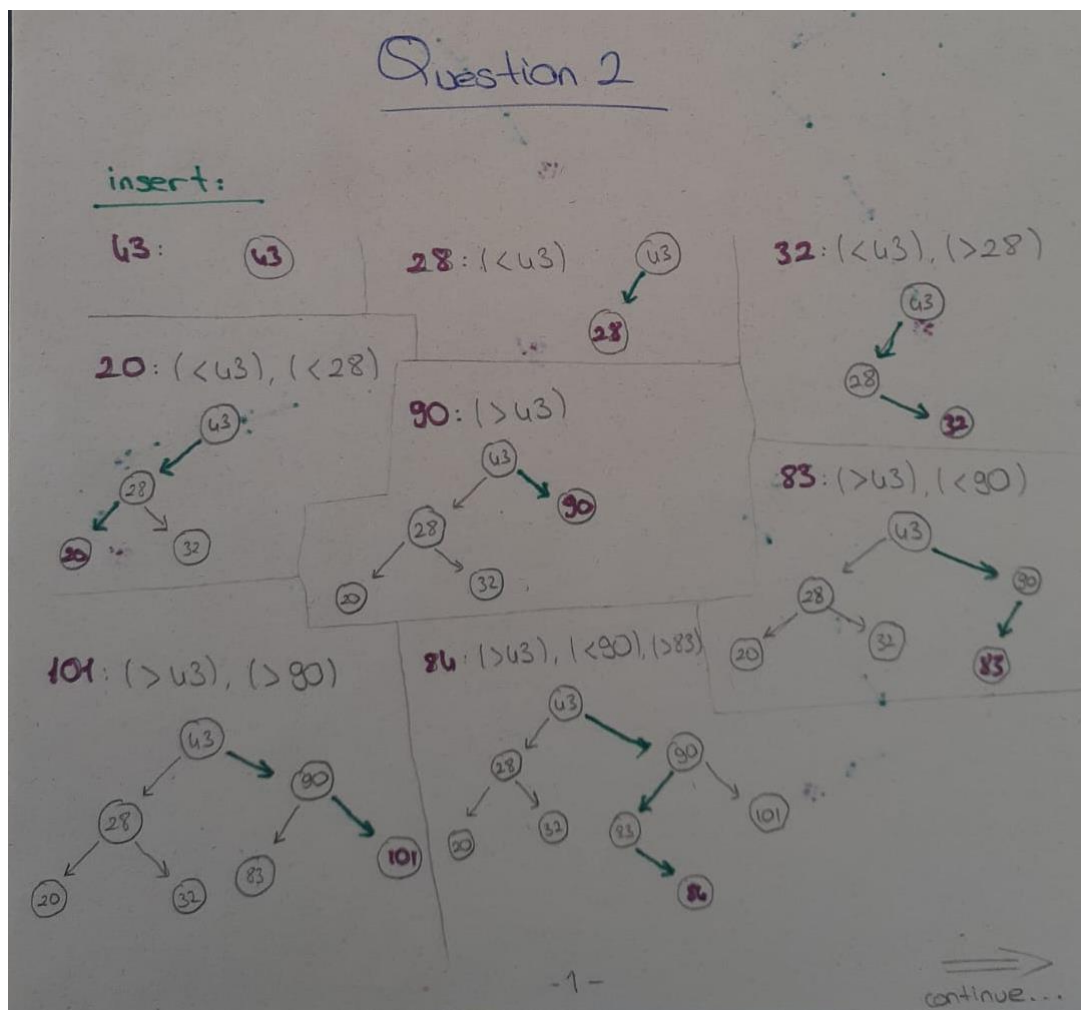
Question 1

Prefix (Preorder Traversal): $- ** AB - + CDE / F + GH$

Infix (Inorder Traversal): $(A * B) * ((C + D) - E) - (F / (G + H))$

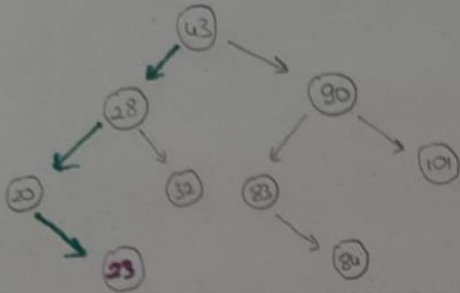
Postfix (Postorder Traversal): $AB * CD + E - * FGH + / -$

Question 2

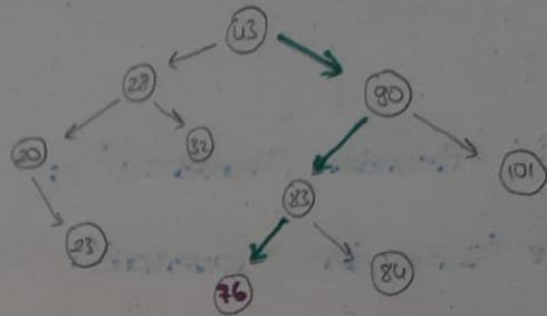


Question 2 continues

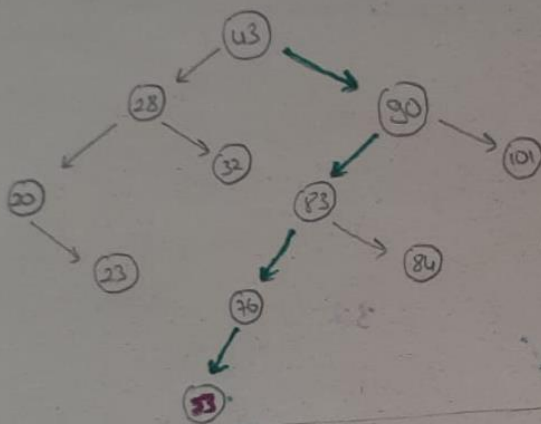
23: (<43), (<28), (>20)



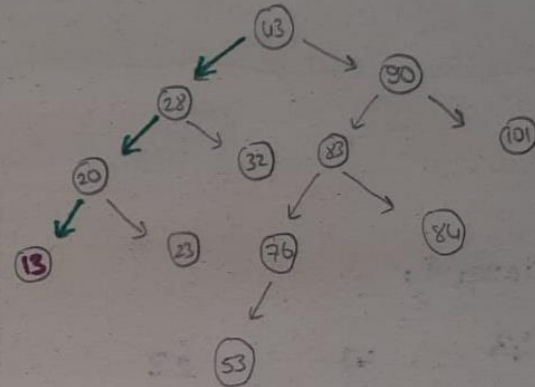
76: (>43), (<90), (<83)



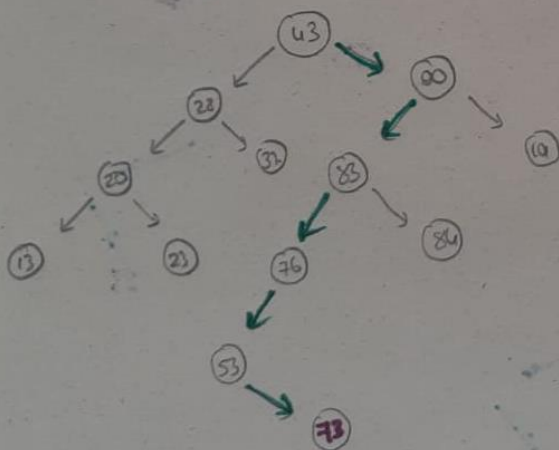
53: (>43), (<90), (<83), (<76)



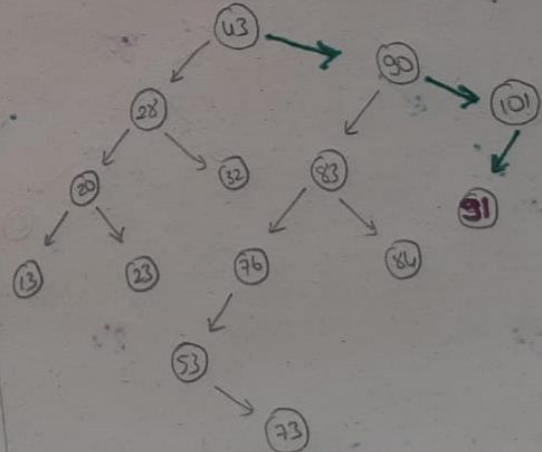
13: (<43), (<28), (<20)



73: (>43), (<90), (<83), (<76), (>53)



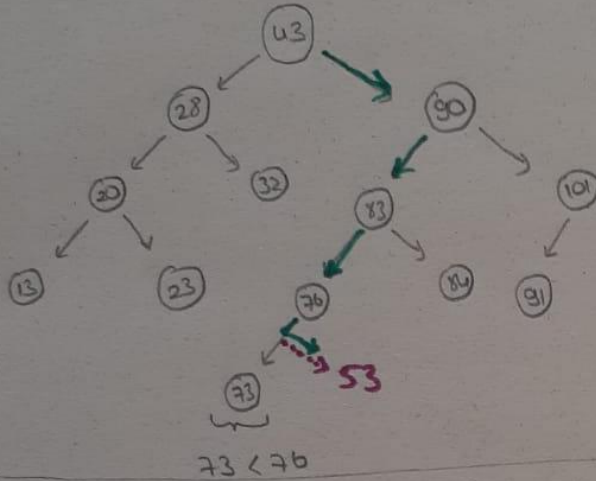
91: (>43), (>90), (<101)



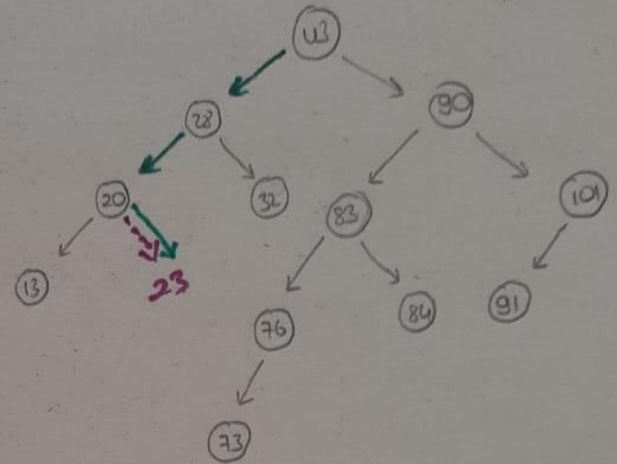
Question 2 continues

delete:

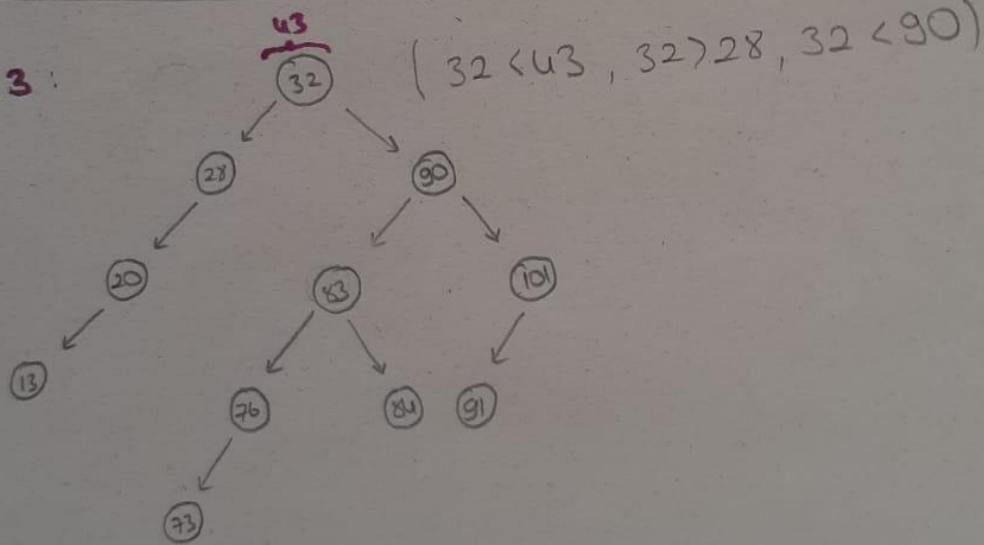
53: (>43), (<90), (<83), (<76)



23: (<43), (<28), (>20)



43:



Question 4

`operator<<(ostream&, const NgramTree&): O(n)`

`operator<<():`

The function calls the 'inOrderTraverse()' function, passing the root node as starting node $\rightarrow O(1)$

The called 'inOrderTraverse()' method performs a traversal where every node is visited only once.

\Rightarrow Worst case: if the tree is skewed, which means each node has only a right child (or only a left child). In such case, the function has to recursively traverse through all the nodes to the left before it can start printing the values $\rightarrow O(n)$

$\Rightarrow O(1) + O(n) \rightarrow \underline{O(n)}$

Question 4 continues

addNgram(const string&): O(n)

addNgram() =

- The first couple of initializations $\rightarrow O(1)$
- Checking whether the tree is empty or not and if empty, creating a node and setting it as root $\rightarrow O(1)$
- If not empty, the code enters a while loop that traverses the tree until it finds the appropriate place for the ngram $\rightarrow O(n)$:
 - If the ngram already exists, increase the count for that ngram $\rightarrow O(1)$
 - If not, create a new node and attach it as a child of 'parent' $\rightarrow O(1)$

\Rightarrow Worst case: the while loop iterate through the whole tree and performs a constant-time action.

$$\Rightarrow 2 \cdot O(1) + O(n) \cdot O(1) \rightarrow \underline{O(n)}$$