2021-2022 Fall Semester

Title: Trees

Author: Alper Mumcular

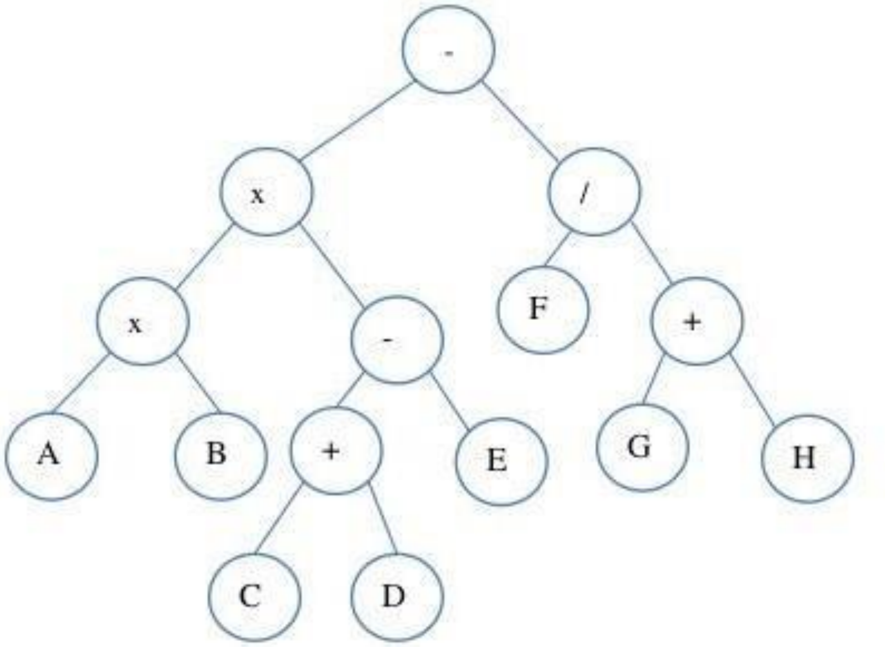
ID: 21902740

Section: 1

Assignment: 2

Description: Report

Question 1



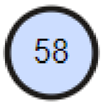
Prefix expression: -xxAB-+CDE/F+GH

Infix expression: AxBxC+D-E-F/G+H

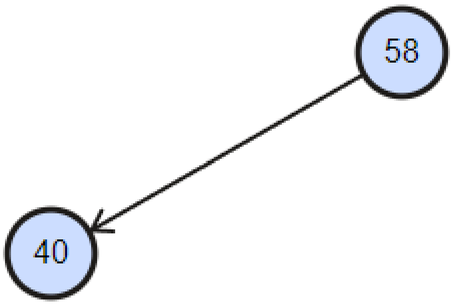
Postfix expression: ABxCD+E-xFGH+/-

Question 2

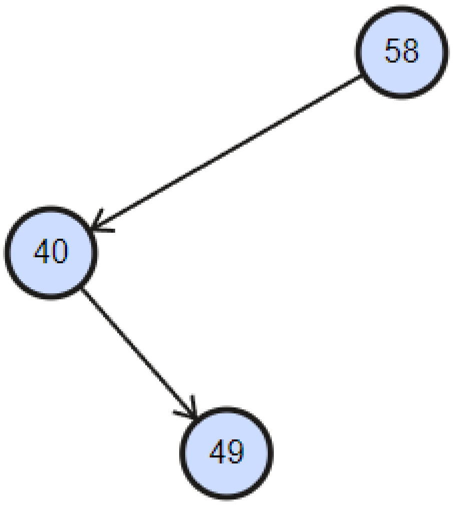
1- Inserting 58



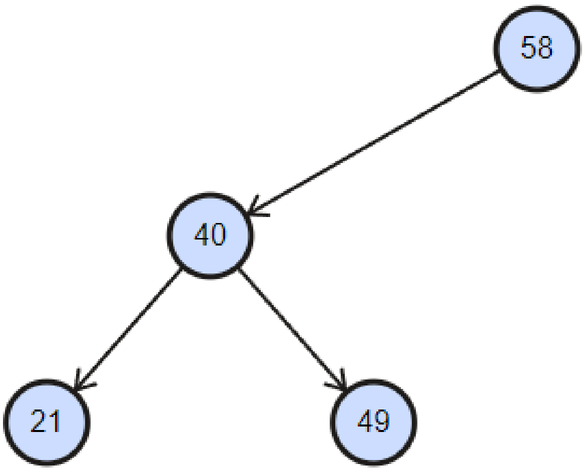
2- Inserting 40



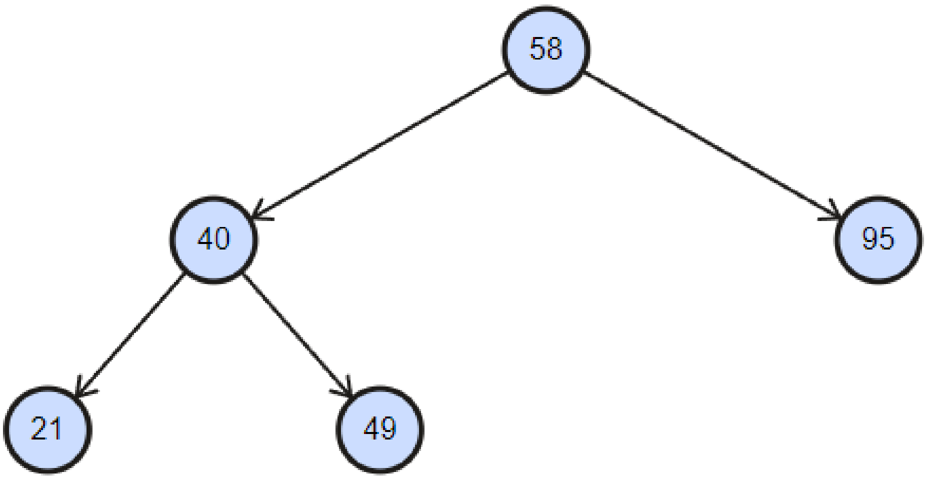
3- Inserting 49



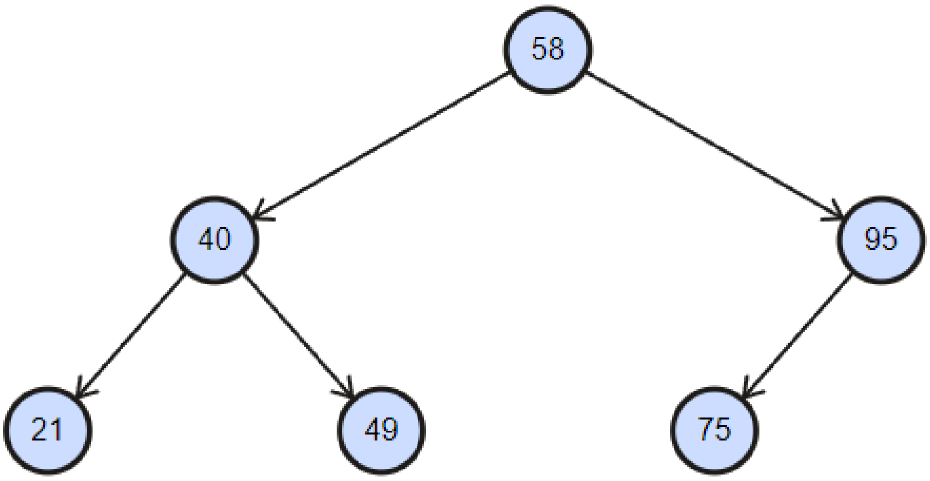
4- Inserting 21



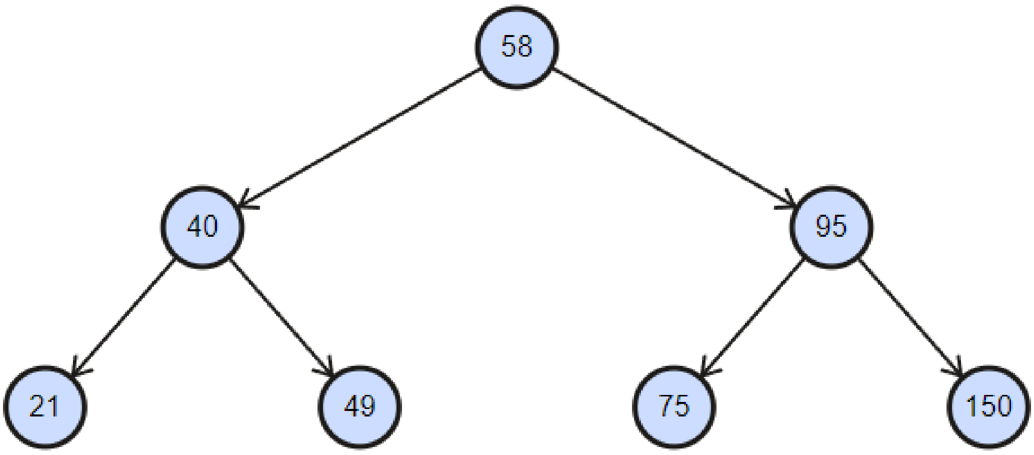
5- Inserting 95



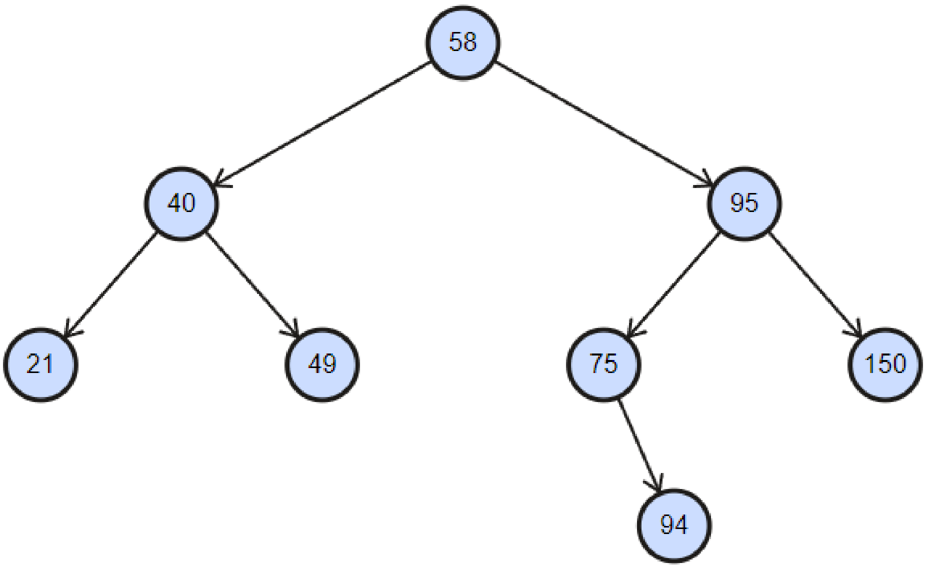
6- Inserting 75



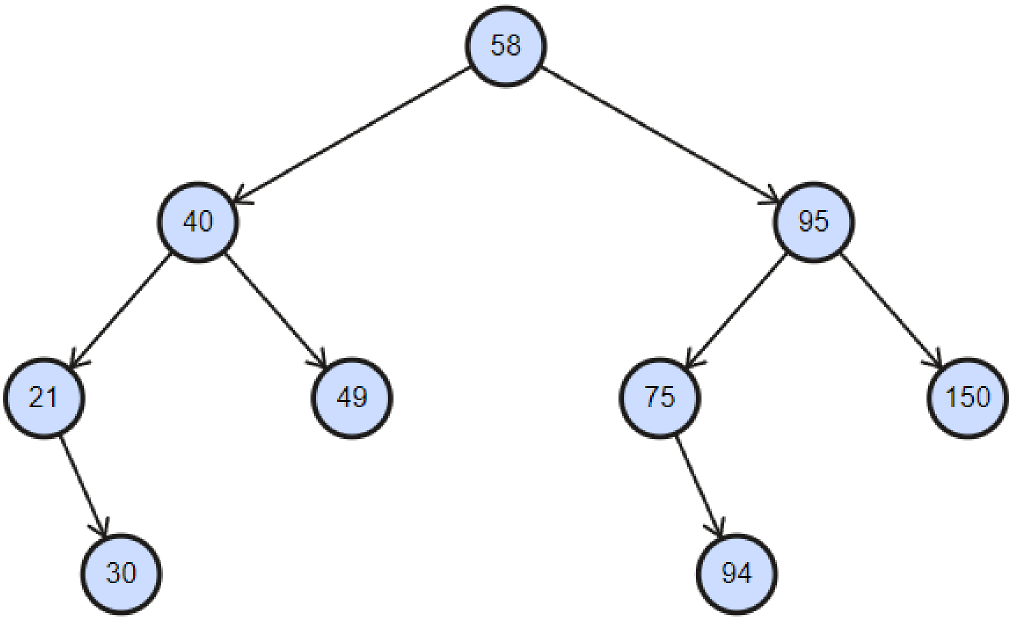
7- Inserting 150



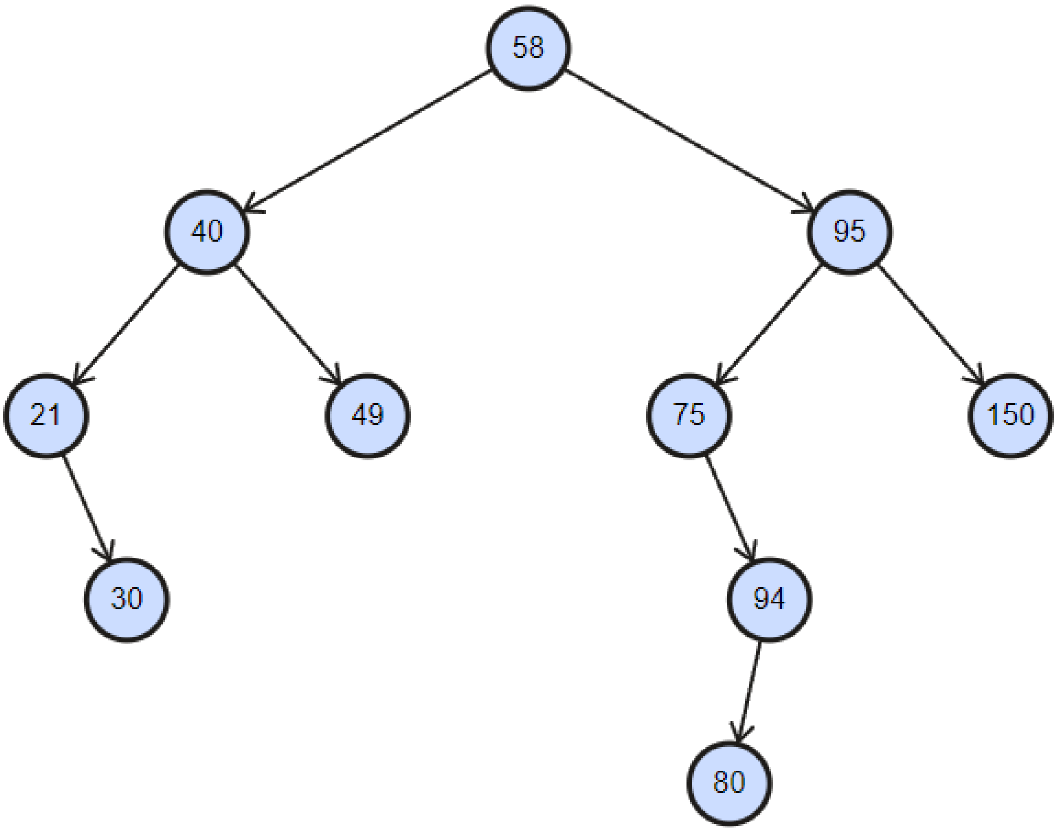
8- Inserting 94



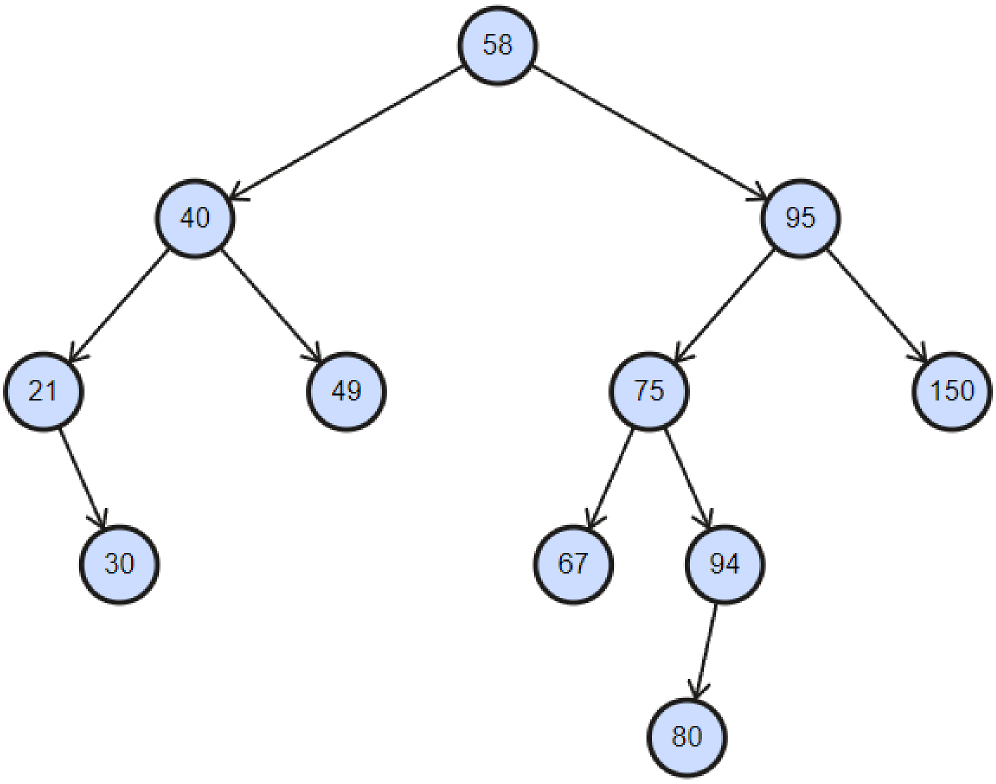
9- Inserting 30



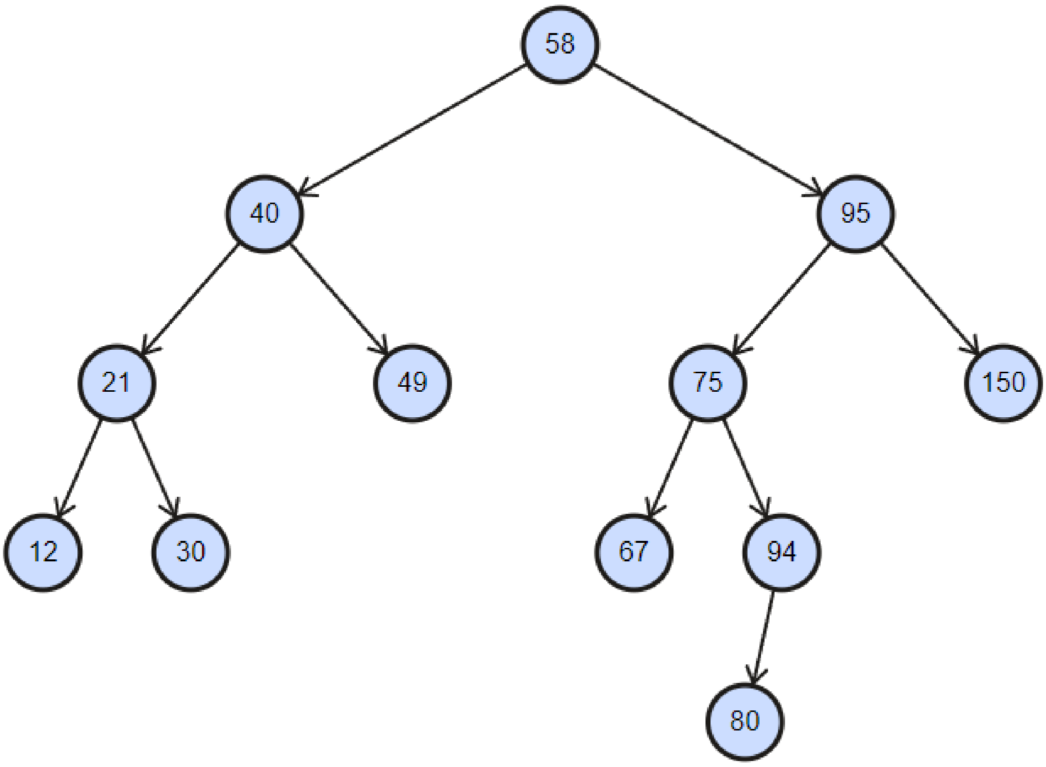
10- Inserting 80



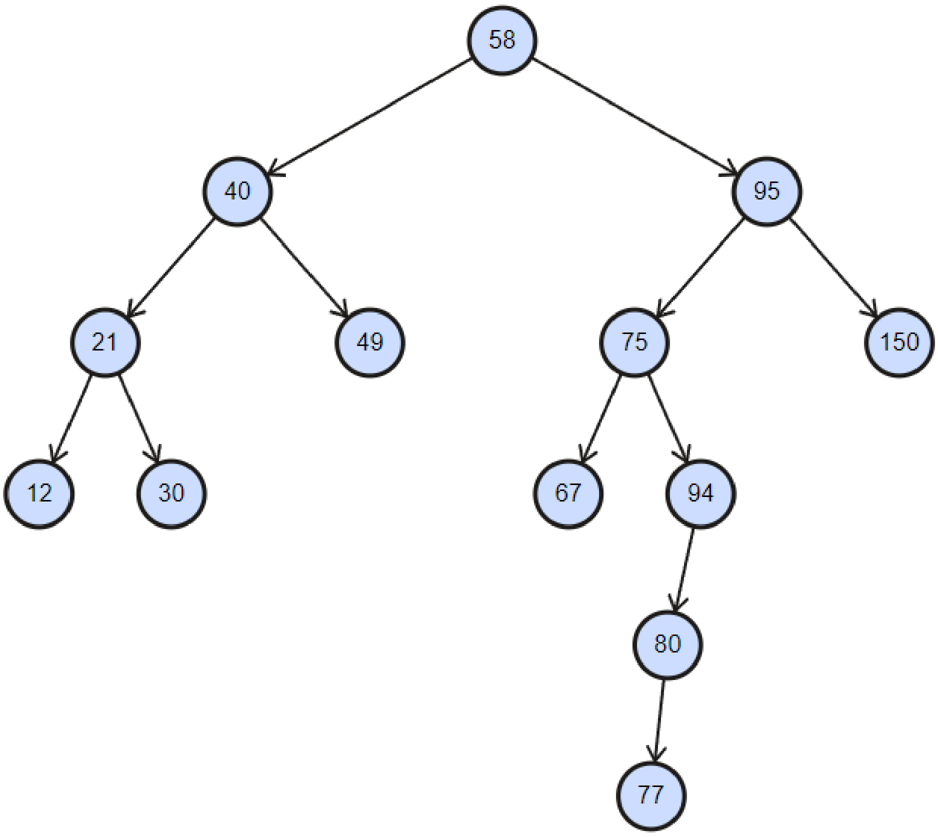
11- Inserting 67



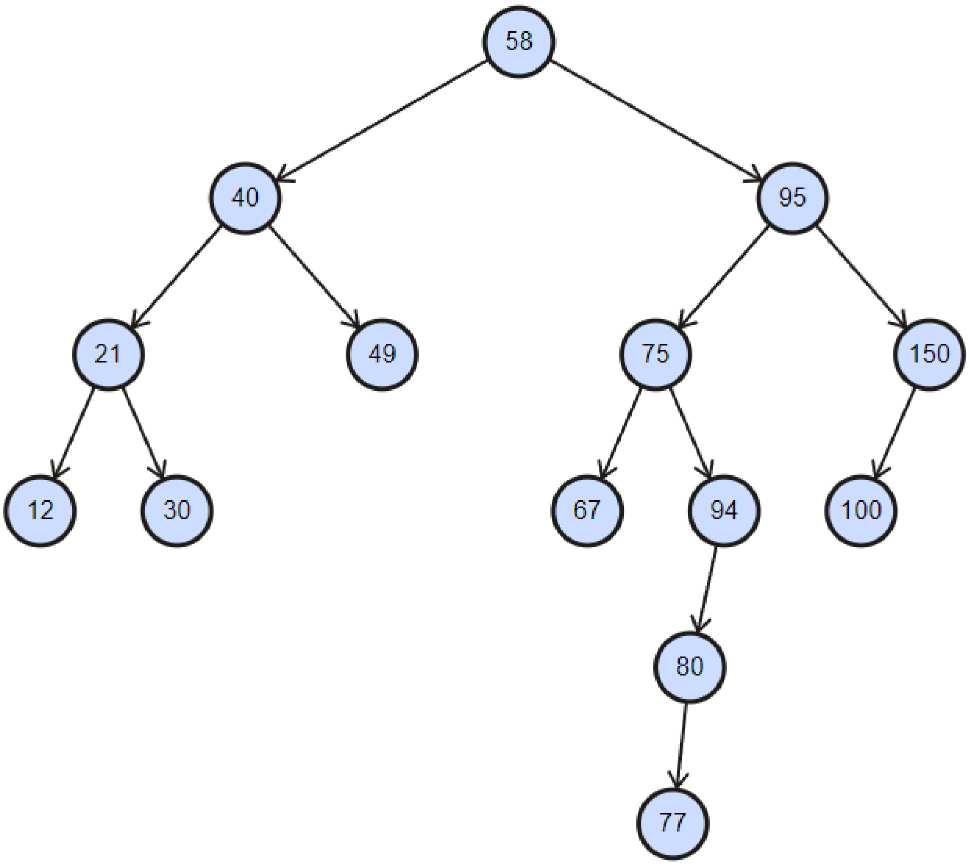
12- Inserting 12



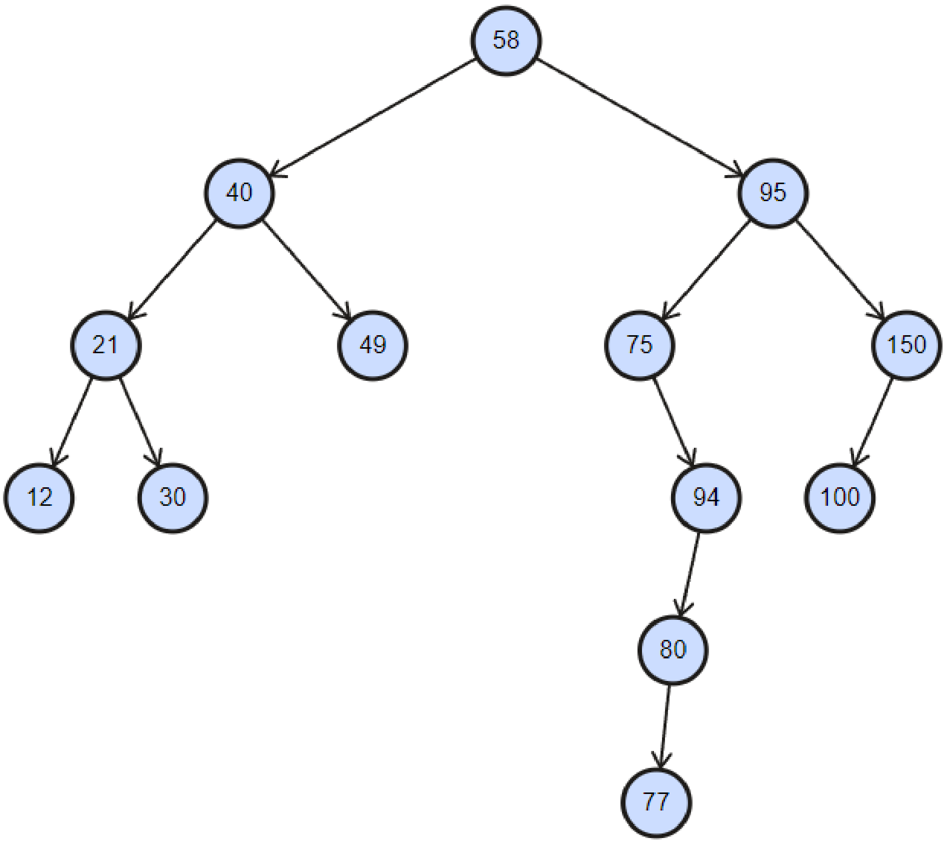
13- Inserting 77



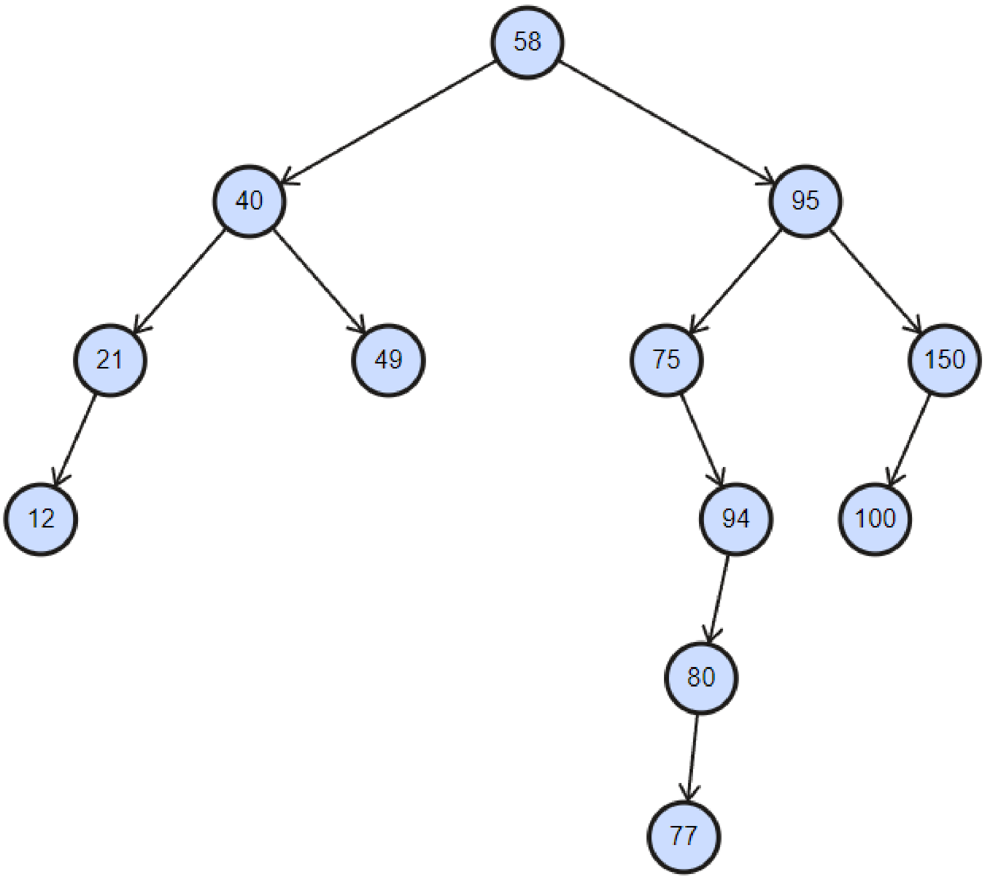
14- Inserting 100



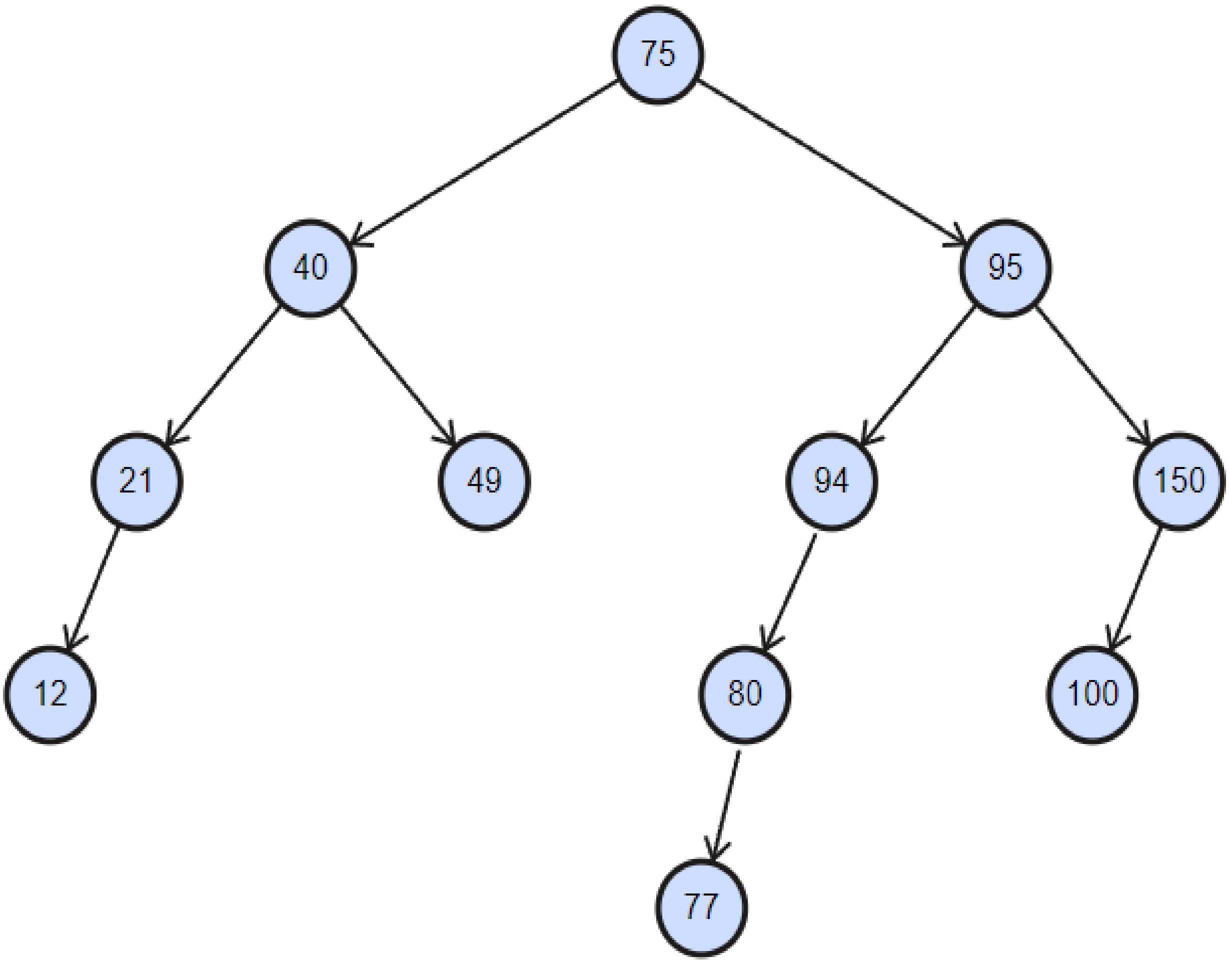
15- Delete 67



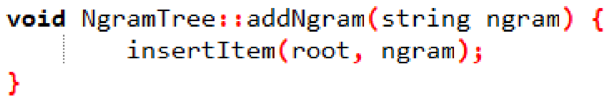
16- Delete 30



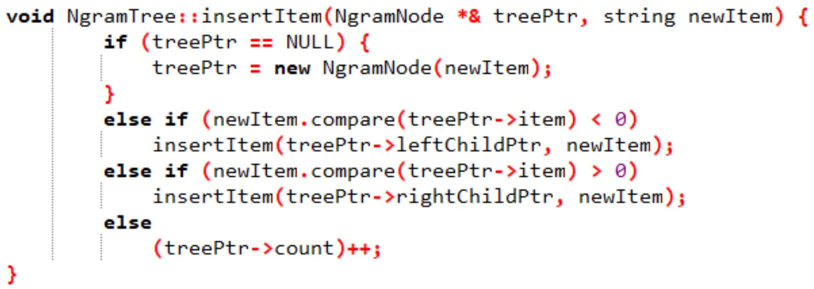
17- Delete 58



Question 4



```
void NgramTree::addNgram(string ngram) {
    insertItem(root, ngram);
}
```



```
void NgramTree::insertItem(NgramNode *& treePtr, string newItem) {
    if (treePtr == NULL) {
        treePtr = new NgramNode(newItem);
    }
    else if (newItem.compare(treePtr->item) < 0)
        insertItem(treePtr->leftChildPtr, newItem);
    else if (newItem.compare(treePtr->item) > 0)
        insertItem(treePtr->rightChildPtr, newItem);
    else
        (treePtr->count)++;
}
```

addNgram: Worst case of addNgram is inserting all items in alphabetical order or reverse alphabetical order in other words the height of the tree equals to size of the tree.

Assume $T(0) = 1$

$$T(n) = T(n-1) + O(1)$$

$$= T(n-2) + O(1) + O(1)$$

$$= T(n-3) + O(1) + O(1) + O(1)$$

…

$= T(n-k) + k*O(1)$ ➔ $k = n$

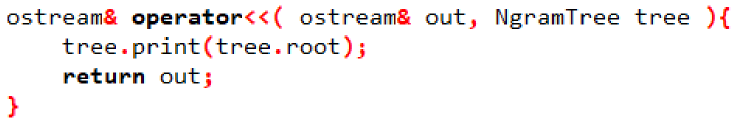
$$= T(n-n) + n*O(1)$$

$$= T(0) + O(n)$$

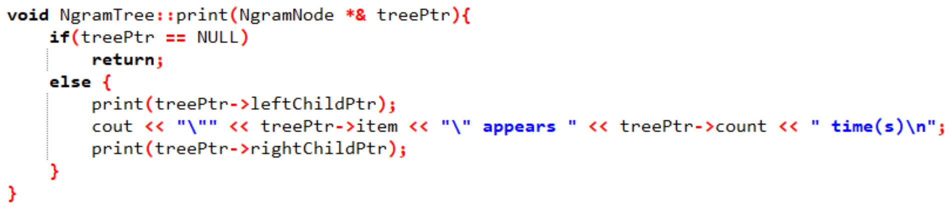
$$= O(1) + O(n)$$

$$= O(n+1)$$

$= O(n)$ is the worst case complexity of addNgram.



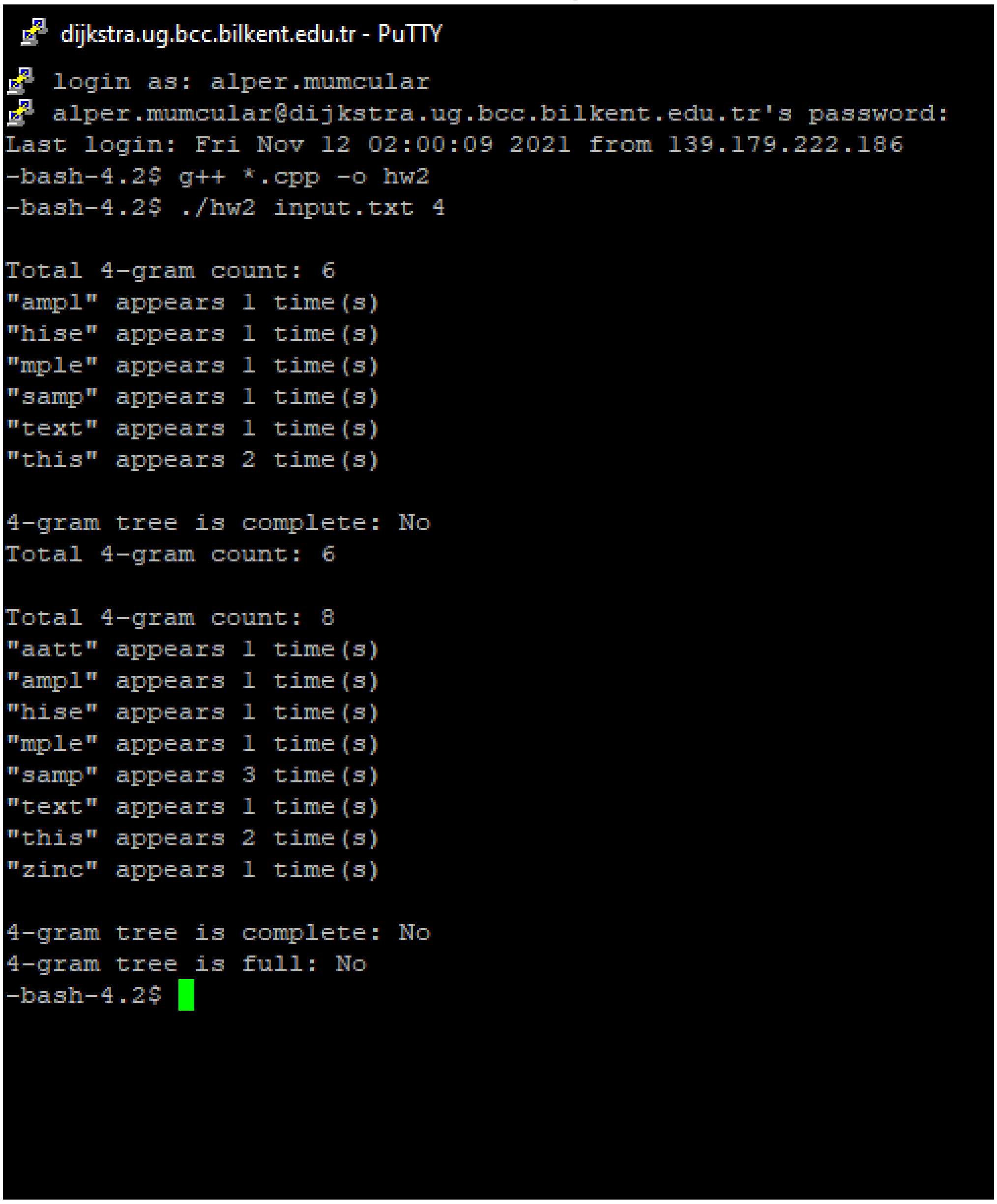
```
ostream& operator<<( ostream& out, NgramTree tree ){
    tree.print(tree.root);
    return out;
}
```



```
void NgramTree::print(NgramNode *& treePtr){
    if(treePtr == NULL)
        return;
    else {
        print(treePtr->leftChildPtr);
        cout << "\"" << treePtr->item << "\" appears " << treePtr->count << " time(s)\n";
        print(treePtr->rightChildPtr);
    }
}
```

operator <<: The worst case of this method is also the best and average case because it has to traverse all nodes alphabetically (inorder). If we assume that we have a NgramTree with size n, operator << method will traverse the all n nodes. Therefore, the worst case of operator << is O(n).

Sample output of the program:



```
login as: alper.mumcular
alper.mumcular@dijkstra.ug.bcc.bilkent.edu.tr's password:
Last login: Fri Nov 12 02:00:09 2021 from 139.179.222.186
-bash-4.2$ g++ *.cpp -o hw2
-bash-4.2$ ./hw2 input.txt 4

Total 4-gram count: 6
"ampl" appears 1 time(s)
"hise" appears 1 time(s)
"mple" appears 1 time(s)
"samp" appears 1 time(s)
"text" appears 1 time(s)
"this" appears 2 time(s)

4-gram tree is complete: No
Total 4-gram count: 6

Total 4-gram count: 8
"aatt" appears 1 time(s)
"ampl" appears 1 time(s)
"hise" appears 1 time(s)
"mple" appears 1 time(s)
"samp" appears 3 time(s)
"text" appears 1 time(s)
"this" appears 2 time(s)
"zinc" appears 1 time(s)

4-gram tree is complete: No
4-gram tree is full: No
-bash-4.2$
```