Sourav Patnaik

2022MT12211

PS11

Q1

[Assignment GitHub Link](https://github.com/sourav977/DSA-assignment)

**Description:**

1. class Dictlist(dict):

is created to Overwrite Dictlist() to store multiple values for same Key

1. class Node:

is Employee class which stores Employee attributes id, name and designation

1. **employees = [ ]** is used to retrieve and store each employee details from input file **inputPS11Q1.txt** and insert into BST by iterating over it.
2. method **insert()** : to insert Employee details into BST
3. method **inorder()**: to store inorder traversal result in inorderTraversal List
4. **inorderTraversal[ ]** List we will use to store inorder traversal result and also for searching employee by name and designation
5. **promptsPS11Q1.txt** file used to read elements and search employee by ID, name and designation and write the result into **outputPS11Q1.txt** file
6. method **searchEmployeeByID():** used to search employee by ID in BST. **empByID = { }** Dictionary is used to store search employee by ID details
7. method **searchEmployeeByNameAndDesignation():** used to search both employee by name and designation. **empByName = Dictlist()** and **empByDesignation = Dictlist()** Dictionary used to store search employee by name and designation details respectively.
8. Output is written into **outputPS11Q1.txt** file
9. File **promptsPS11Q1.txt** contains the items to search

**Time Complexity:**

**For the python code that I have written**

1. For **Insert()** method:

For inserting element , we have to traverse all elements. Therefore, insertion in binary tree has worst case complexity of O(n), In average case, is O(log n)

1. For **inorder()** method:

we have to traverse all elements. Therefore, inorder traversal in binary tree has worst case complexity of O(n), In average case, is O(log n)

1. For **inorderTraversal[ ]** List

Append element to list will cause O(1)

1. For **searchEmployeeByNameAndDesignation() method:**

We have to traverse/search whole **inorderTraversal[ ]** List, if match found, store them into **empByName** and **empByDesignation** Dictionary. When storing elements to dictionary for both average case and worse case, it is O(N). when finding values for corresponding keys in the dictionary, it is O(1)