

```

#
# Machine Problem - 7
#
# Aravind Kumar Kaspe
# Banner ID: 001291145
#
# Description : The program involves creating three data structures: an
#               Unsorted List, a Sorted List, and a Binary Search Tree.
#               It generates 10 random integer values ranging from 1 to 150
#               and inserts each value into all three data structures.
#               To display the contents, it utilizes different methods for
#               each structure: the print() function for the Unsorted List,
#               the printList() instance method from Lab 11 for the Sorted
#               List, and the inorder() method for the Binary Search Tree,
#               which performs an inorder traversal to print its contents.
#
#
from SortedList import *
from UnsortedList import *
from BinarySearchTree import *
from random import randint

def trial(storage, highest, numSearches):
    #
    # Conducts multiple searches in the data structure defined by the parameter
    # storage. Generates a random int value between 1 and highest. Uses the
    # instance method storage.find() to search the data structure for that int
    # value. Repeats this process numSearches times. Counts the number of
    # successful searches, the number of unsuccessful searches, the number of
    # total visits for the successful searches, and the number of total visits
    # for the unsuccessful searches. Returns those values in a list.
    #
    # storage The data structure to be searched. The data structure
    # must store int values and have an instance method find()
    # that conducts the search and returns a list where the
    # first value is True if the search was successful, False
    # otherwise, and the second value is the number of visits
    # made during the search.
    # highest An int value between 1 and highest will be randomly
    # generated and searched for.
    # numSearches The function will conduct this many searches.
    #
    # Returns a list with 4 values... the number of successful searches, the
    # number of total visits for those successful searches, the number of
    # unsuccessful searches, and the number of total visits for those
    # unsuccessful searches.
    #
    successfulSearches, totalSuccessVisits, failureSearches, totalFailureVisits =

```

0, 0, 0, 0

```
for search in range(numSearches):
    randomNum = randint(1, highest)
    found, visits = storage.find(randomNum)

    if found:
        successfulSearches += 1
        totalSuccessVisits += visits
    else:
        failureSearches += 1
        totalFailureVisits += visits

return successfulSearches, totalSuccessVisits, failureSearches,
totalFailureVisits

if __name__ == "__main__":
    unsortedList = UnsortedList()
    sortedList = SortedList()
    binarySearchTree = LinkedBinarySearchTree()

    for i in range(10):
        randomNum = randint(1, 150)

        unsortedList.append(randomNum)
        sortedList.insert(randomNum)
        binarySearchTree.insert(randomNum)

    print(f'Unsorted List: {unsortedList}')
    print(f'Sorted List: ', end='')
    sortedList.printList()
    print(f'BST Inorder Traversal: ', end='')
    binarySearchTree.inorder()

    unsortedList = UnsortedList()
    sortedList = SortedList()
    binarySearchTree = LinkedBinarySearchTree()

    for i in range(100):
        randomNum = randint(1, 150)

        unsortedList.append(randomNum)
        sortedList.insert(randomNum)
        binarySearchTree.insert(randomNum)

    print()

    print(f'\n{" ":>22}{"Found":>8} {"Avg Visits":>10} {"Not Found":>10} {"Avg
Visits":>10}')
```

```
trailResult = trial(unsortedList, 150, 100000)
print(f'Unsorted List {" ":>8}{trailResult[0]:>8}
{trailResult[1]/trailResult[0]:>10.2f} '
      f'{trailResult[2]:>10} {trailResult[3]/trailResult[2]:>10.2f}')

trailResult = trial(sortedList, 150, 100000)
print(f'Sorted List {" ":>10}{trailResult[0]:>8}
{trailResult[1]/trailResult[0]:>10.2f} '
      f'{trailResult[2]:>10} {trailResult[3]/trailResult[2]:>10.2f}')

trailResult = trial(binarySearchTree, 150, 100000)
print(f'Binary Search Tree {" ":>3}{trailResult[0]:>8}
{trailResult[1]/trailResult[0]:>10.2f} '
      f'{trailResult[2]:>10} {trailResult[3]/trailResult[2]:>10.2f}')
```