Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

1. Introduction

The purpose of this lab is to :

a. Study and create generic classes.
b. Study and implement interfaces
c. Study and implement singly-linked non-circular linked lists
d. Study and implement iterators

First goal was to create a Node class for the implementation of the linked list with an integer key value, a generic V class value and a Node pointer next with points to its next Node.

Second goal was to create a MyLinkedList class which implements Iterable and has two pointers:

a. Head pointer pointing to first Node in list
b. Tail pointer pointing to last Node in list.

MyLinkedList class also has an int value length which recorded the length of the list.

Third goal was to create a MyLinkedListIterator which implemented Iterator

2. Approach

The design of Node class is simple. Getters and setters were created to obtain and set respectively, values, keys and next pointer objects.

The design of MyLinkedList class revolves around one paramount rule: the head, tail or any node within the linked list, should not be accessed outside the list with methods. If this functionality is enabled, users of MyLinkedList class can manipulate the linkage between nodes and thus, may result in errors if not careful. Thus access to values is possible with the use of indexes or keys of integer data type. Values can be added only to the beginning and end of list, per instruction of the lab manual instruction. Values at an index can also be removed.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

The design of MyLinkedListIterator class is somewhat confusing. Its instantiation required a Node index and the object to be iterated itself. MyLinkedListIterator design made it difficult, if not possible, to instantiate it outside MyLinkedList.

The reason is that no nodes can be accessed from outside MyLinkedList and thus, if a class wants to access an index in MyLinkedList, it needs to be created within it.

Thus test methods of MyLinkedListIterator such as next(), hasNext() and remove() were tested in MyLinkedList test class for this reason.

3. Methods

Runs are not hard coded into program and thus, a choice of the user.

Parameters used by the program, as instructed in the lab manual, is the file name for data written by program.

NumberOfItems data to control the number of data inputs to a MyListStringContainer objects and number of items through which to search.

4. Data and Analysis

Number of runs for this data is 1000.

Number of items sorted and added to StringContainer objects are 2000 to 9000 inclusive.

RandomStringGenerator was imported from previous Lab (Lab 4)

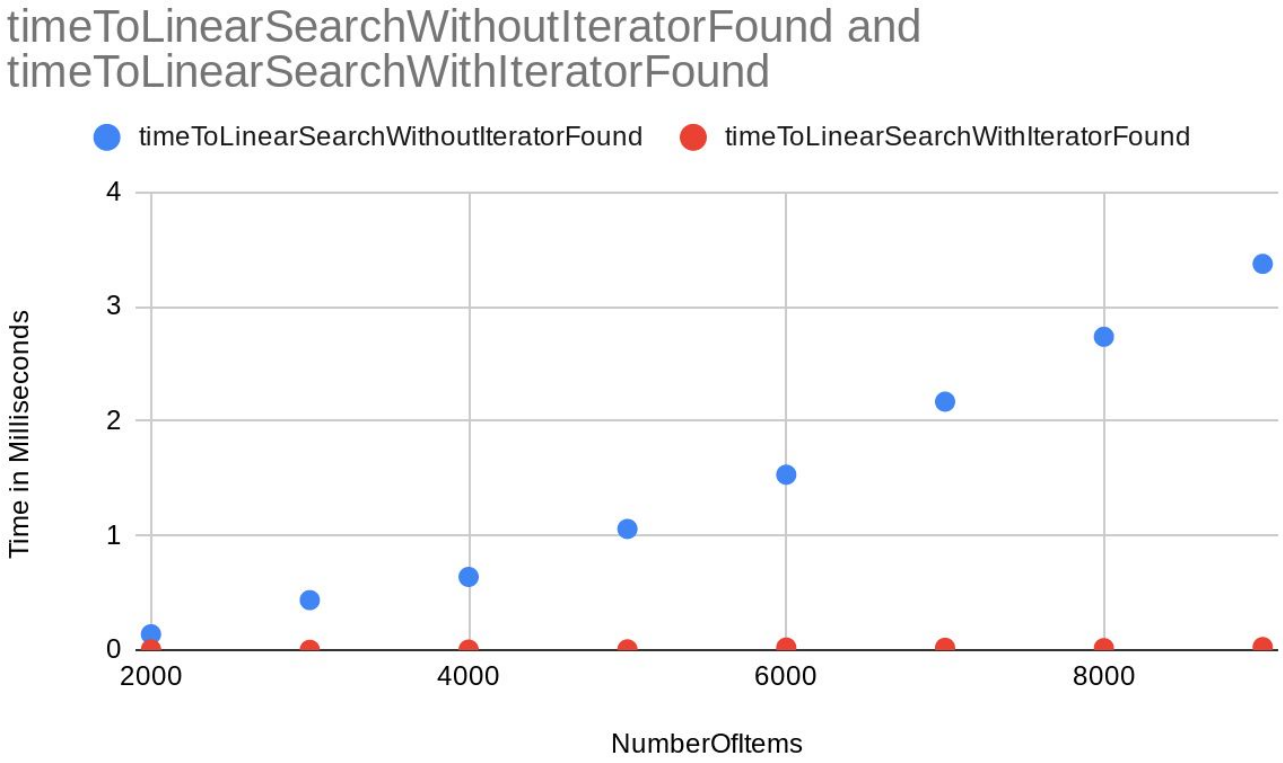
Seed used for RandomStringGenerator was an arbitrary number of int 1000 and strLength of 5 for RandomStringInteger class.

Data generated is attached as a .csv file with name “output09282019.csv”.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

Searches

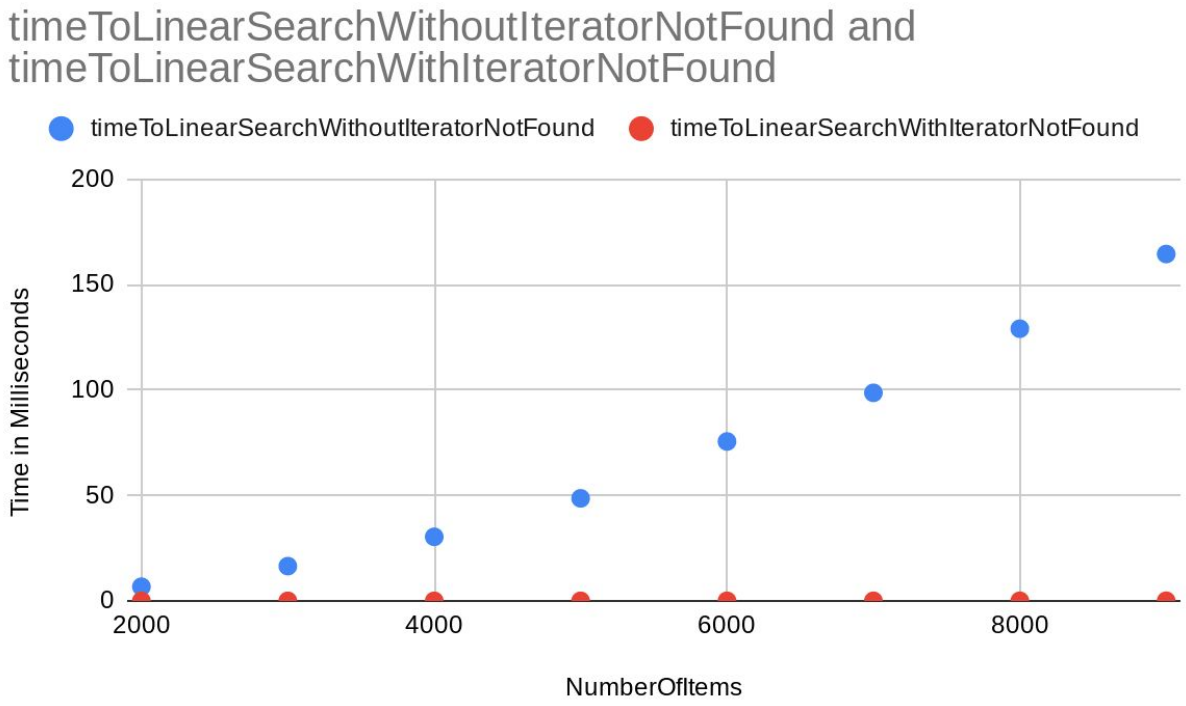
LinearSearch() and LinearSearchIterator() for a String key in the list



The time taken to find an item in MyLinkedList with no iterator increased as numberOfItems increased but the time to do a linear search to find an item in My LinkedList stayed around the same value as numberOfItems increased.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

LinearSearch and LinearSearchIterator() for a String key not in the list



Similar trend to the first graph is also noticed. The time taken to find an item not in MyLinkedList with no iterator increased as numberOfItems increased but the time to do a linear search to find an item not in My LinkedList stayed around the same value as numberOfItems increased.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

5. Conclusion

All classes created performed as they should per lab manual instruction.
Iterators are faster in accessing nodes compared no non iterators.
Major difficulty of the lab was with the re-assignment of keys or nodes of indexes has met in this lab.

6. Reference

Java Iterator and Iterable documentation were referenced when completing this lab.