Fiona Liu

Fxl154

Final Project: Progress Report

1. IntArrayBag and IntLinkedBag data structures from HW3 were modified to be sorted in ascending order. The run times of 3 methods were analyzed – add, addAll, and remove. The procedure involved rewriting the add method to include sorting the target element into ascending order. The addAll method is then modified to implement the merge sort algorithm. The remove method implementation is the same from that of HW3. A main method is added to the IntArrayBag and IntLinkedBag class to demonstrate the implementation of these revised methods. The run times of each method are recorded, and then graphed with respect to its add, remove, and addAll implementation of a bag of length N. In our demonstration, N increments to 5000.
2. The following first describes the time complexity, then the results of the analysis.
   1. The add method had the same implementation for both data structures. The add method traverses through a string of some length, then places a target element in order.

For IntArrayBag, the time complexity is O(N) because the method needs to traverse through some length of an array before pushing the other elements to the next position.

For IntLinkedBag, the time complexity is O(N) because the method needs to find the value in the bag that is greater than the target element, which could be all the elements in the bag.

The following is a run time comparison between the two classes for the add method.

* 1. The addAll method is modified to use the Merge Sort algorithm. The existing algorithm completed in HW3 is used, followed by a Merge Sort implementation to sort the new array in order.

For IntArrayBag, the time complexity is O[(N1+N2)log(N1+N2)] where N1 elements are added to the list and N2 elements are already in the bag. N1 elements are first added to the end of the bag, before merge sort is called to sort the final array of (N1 + N2).

For IntLinkedList, the time complexity is O[([(N1+N2)log(N1+N2)] as the linked list elements (N1) are added to the start of the array. Similarly, merge sort is called to sort the final array.

However, there is a Stack Overflow exception when using merge sort with IntLinkedBag for N>3000, a result of there being too many recursive calls. A solution for this is to alternatively call the add() method for every element in list N1, or rewrite the merge sort algorithm without recursion.

Below is the run time analysis for the two classes.

* 1. The remove method follows the same implementation from HW3, where the method traverses the bag until it finds the target element to be removed.

For IntArrayBag, the time complexity is O(N) because the element is found, removed, and then all the elements before it are shifted back one position.

For IntLinkedBag, the time complexity is O(N) because the element is found, removed, and then all the elements before it are shifted one position.

Below is the run time analysis for the two classes.

1. The purpose of this assignment is to compare the run times of two similar, but different data structures. Implementing linked lists is often preferred over implanting arrays because array lists have a predefined number of arrays in a lists prior to implementation. Linked lists, however, are free to be added on. Performing this run time analysis is a point of interest because although in many situations, linked lists seem more logical, they are ultimately slower than an array list.
2. There have been no significant problems while working as a team. All material related to this assignment have been archived in a Github repository to ensure ease of collaboration and storage of previous versions.

A problem with the code is a Stack Overflow exception when IntLinkedBag runs merge sort algorithm for N greater than 3000. This is because there are too many recursive calls, which Java is unable to handle. Through a brief Google search, it was recommended to consider implementation of merge sort for Linked Bag without recursion. Our next steps will be to experiment with merge sort using either a queue or two stacks.

1. Varun worked on the add, addAll, and remove methods for IntLinkedBag. Varun wrote the presentation. Fiona worked on the add, addAll, and remove methods for IntLinkedList. Fiona wrote the progress report.