Assignment #1: Project Overview

The Objective of this project is to modify the ArrayList class, provided by Weiss, ensuring items are added in a collection in ascending order and that this order is maintained throughout subsequent operations. Considering this objective, there are two main features that are crucial to maintaining the order of each collection: an efficient search algorithm and a shifting algorithm.

The first thing I created in this project was the search algorithm ‘appropriateIndex()’. It determines the correct placement of an item in a sorted list. This method uses Binary Search with a time complexity of $O(log\ N)$, compared to the time complexity of linear search $O(N)$. This method of search is made separate from the ‘binSearch()’ method because appropriateIndex() will always return an index while ‘binSearch()’ will return -1 if an item is not found.

For shifting algorithms, I designed two separate implementations. The first way appears in the ‘add()’ method. Starting from the end of the collection, each element is moved one index to the right until ‘appropriateIndex()’ is reached where the new item is inserted. The second shifting algorithm was implemented in the ‘set()’ method. This design depends on whether ‘appropriateIndex()’ is greater than or less than the target index. If the target index is greater than the appropriate index then elements are shifted right similar to the ‘add()’ method. Otherwise, the elements are shifted left until the ‘appropriateIndex()’ is found.

For the Result class, I implemented a single pass technique with a time complexity of $O(N)$ instead of a nested loop of $O(N^2)$. This method only works if the collection is sorted. Within the implementation of Result, I created a method that would calculate the Mode and the Count and assign them to their respective private variables. Whenever the methods ‘mode()’ and ‘count()’ are called the mode and count of the collection is recalculated. I decided on this because it is an optimized way of getting a collection’s mode and count in a changing, sorted list.

The objective of this project is to modify the ArrayList class, provided by Weiss, to ensure that items are added to a collection in ascending order and that this order is maintained throughout subsequent operations. In light of this objective, two main features are crucial for maintaining the order of each collection: an efficient search algorithm and a shifting algorithm.