Project 1 – Write-up

CS249

Brandon Horner

September 26, 2014

**Executing the code:**

Part I: A, can be found in the BubbleSort.java, InsertionSort.java, SelectionSort.java, these contain the three simple sorts that we went over in class. I tested each sort with BubbleSortApp.java, InsertionSortApp.java, SelectionSortApp.java, which each create differently sorted arrays, sort each and display the comparisons/copies totals performed in the sorts.

Part II: A, can be found in Queue.java, partnered with the QueueDisplay.java for testing. B can be found in the Dequeue.class, however I did not understand part II: B so I do not have a working display method for said class.

**Big(O) Efficiency:**

In the InsertionSort.java, insertSort() has a big(O) efficiency of O(n^2) due to its nested loops. Upon adding the comparisons and copies variables into insertSort(), I could see that the efficiency of this algorithm becomes much better the closer to sorted the list is (for example, the reverse ordered array had 45 copies and the almost sorted array only had 8). However it does also verify the O(n^2) efficiency.

BubbleSort.java’s Bubble() method also has a big(O) efficiency of O(n^2). The actual efficiency is something like O(n^2)\*2 because there are two loops nested inside a while loop.

Finally in the SelectionSort.java file, selectionSort() is O(n^2) as well, because of the nested loops. Copies became significantly more the less sorted the original array was.

The Queue.java has a method called queueDisplay() with big(O) efficiency of O(n) because there is only one loop to go through each time.

**Further comments:**

In the classes that test the sorts, I show multiple arrays, with variable “sortedness”, and the number of comparisons and copies done in each method. The comparisons for each sort were always N or less.

My QueueDisplay.java class initializes three different arrays: an empty one, a single element array, and one with five elements that I insert, remove and mess around with. I display the result after each transformation.

I gave my best shot at Dequeue but do not fully understand how it worked.

\*I did not collaborate with anyone.