

# Project 1: Comparing Algorithms

Steven Turner N00836867

Monday October 24, 2016

## 1 Description

The two classes I used were an array-based Queue for DS1 and an Ordered Linked List for DS2. The Queue was taken from the Data Structures textbook that we are using for this semester and the Ordered Linked List was one that I partially created last semester. I say partial because part of the code was written originally by Professor Graham.

## 2 Analysis of Expected Run Time

I expect the run time for worst case scenario for the Queue to be  $O(1)$  and Doubly Linked List to be  $O(n)$ . I believe that this is true in the case of the Queue because to add an item, you just go to the next spot in the array and if it is true, add the item and if not, throw an exception. And likewise with remove, you go to the last spot in the array and just remove it.

However with an Ordered Doubly Linked List, you have to traverse every item in the list until you either have found a spot where the item you are inserting would best fit, or have reached the end of the list. And likewise since you are removing from an ordered list without any parameter, you would just remove from the end of the list which would mean traversing through every object until you reach the end and then remove it.

## 3 Experiment Ran

The experiment that I ran was I counted all the primitive operations for the insertion and deletion of an element in both the Queue and the Linked List when I added and deleted 1 million elements. I noticed that for the Queue, it ran in a constant time and that for the Linked List, it ran in  $O(n * c)$  time.

## 4 Analysis of Potential Switches to Implementations

I believe that the purpose of a Queue and an Ordered Linked List are two different things. Would it be better to use an array for a Ordered Linked List? Only when it comes to adding and inserting at the end is it better to use an array. If you were to add in a sorted list or add to the beginning of the list, it is better to use a linked system. And that's the purpose of a Ordered Linked List, to add in an ordered fashion.

## 5 Debugging and Difficulties

I didn't really run into too many bugs once I found my Data Structures. One thing that I found difficult however, is how to compare data in a sorted list if it is of generic type T, however a look at my notes from previous classes reminded me of that T would have to extend from Comparable<T>, so I can use the compareTo method.