CPS 151/Spring 2017 Assignment 4 Selection sort and merge sort for a subclass of a queue of integers
Assigned: 24 April Due: May 1, 11 pm on Isidore Cut-off: May 4, 11 pm on Isidore

In class Merge Sort was discussed for an array. The original array and the subarrays that are created are accessed only sequentially. That means Merge Sort can be easily implemented for a collection that only allows sequential access (such as a linked list or even a queue). **The random access capability provided by an array is not required to implement Merge Sort.**

In this assignment, we will extend the IntQueue class to create a class that has additional methods, so that a client can implement Merge Sort and selection sort for the collection.

Examine the existing code and fill in the missing pieces.

The client should be modified so that it shows that sorting works for a small random queue. Then it should create random queues of large sizes (say 10,000 to 100,000 in steps of 10, 000). The client should produce a table of queue sizes and corresponding sorting times in milliseconds. The client should also get the times for sorting random queues using selection sort for the same sizes and include that for comparison.

Quite a bit of startup code is provided which was discussed in class.

These are the results I got from my program. Different runs do tend to give different results.