Diego Ochoa

CS 2235 Kerby

9/28/21

**Homework 4**

Array time is roughly equal to O(n \* 1.5)

DLL time is roughly equal to O(n \* log n)

It seems as though, for this application, the implemented array was faster than the Doubly Linked List, which is speculated to be due to a lack of optimization. In the theory, the DLL code would have performed better than the Array code. However, not only the iteration, but the implementation of an array was vastly quicker than that of the DLL, whereas the total DLL time increased dramatically as the list size grew.

I had predicted that the Array iteration may be quicker than the DLL iteration (due to the fact that adding to the array would most likely be an O(1) process and adding to a DLL requires a lot of circulation and invocations), and that the DLL initialization would be quicker than the array initialization (given the DLL’s theoretically quicker method of initialization requiring less circulation than an array would.) However, regardless of the list size, even the DLL’s initialization was slower.

This was then tested on both a 1-dimensional array, as well as a 2d array, however, even then, the array was quicker. The reason this leads me to speculate that this is due to poor optimization is that: in theory, the linked list should initialize quicker but, in this case, both the initialization and random parsing of the DLL required the invocation of multiple methods whereas arrays generally only required a for-loop or a nested for-loop.