Exploration: Dequeues

1. What are the similarities and differences between a deque, a queue, a stack?

Similarities are that they are all and ordered collection of items where the items remain positioned in the collection. Differences are that a queue has two ends where adding and removing can occur and in no particular order. It is up to the programmer to decide the order but it can be inserted and removed from either side of the ordered collection. Stack has a LIFO order and Queue has a FIFO order.

1. What are the performance implications of implementing a deque using an array? Explain each operation and its corresponding time complexity including the why for each operation.

Depending on the implementation adding and removing (pop(), addFront(item), addRear(item) and the operation chosen it will be either O(1) or O(n). Adding to the front or rear will be O(1) because it always leads to one place either the front or the rear or so I thought. However as you add to an array or subtract from an array the number of variables in it changes but the size of the array remains the same unless you make changes to that as well, pop() for instance would depend on the order in which you implemented. Since a Deque does not have any rules surrounding order like FIFO or LIFO. I do know that I’s important to know where the front and where the rear of the array are located and where you’re trying to input values.

1. Big-O tables list worst case insertion and deletion for deque's as O(1). How is that possible?

I’m honestly not sure, I thought that deleting and insertion was possibly O(n) depending on where you inserted or had to delete from.