

Timothy Ha
1367917
junkwan
CSE 373 HW 1
04.08.15

1. To test if my ArrayQueue and ListQueue implementation were correct I created public methods in each class such as a toString method and for the ArrayQueue a getFront and getBack method. The toString method is useful because it helps to visual the entire queue for either the ArrayQueue or ListQueue then I can print it out every time I enqueue or dequeue to make sure those two functionalities are working. The getFront and getBack methods for the ArrayQueue return the positions of the front and back of the ACTUAL queue, since it's a circular array, then I'm able to compare what I worked out on paper to the returned positions. For each class I tested the ends to make sure there is an error message when an empty queue is dequeued or when a full queue is enqueued.
2. 128 is 2^7
1 million is roughly 2^{20} , 1 billion roughly 2^{30} , 1 trillion roughly 2^{40}
To resize for adding 1 million items, I would need to double the original $(20-7) = 13$ times.
1 billion items requires doubling $(30 - 7) = 23$ times
1 trillion items requires doubling $(40 - 7) = 33$ times
3. I would use two stacks. The first stack, stackIn would look just like a queue and I would push items onto it. If I wanted to dequeue the first(bottom) item, then I would pop/push every item into my second stack stackOut. Then if I wanted to dequeue again I can pop off of stackOut until it is empty, and if I want to add items I can push onto stackIn. Assuming stackIn has items in it the pseudocode is:
while (stackIn is not empty AND stackOut is empty)
 stackOut.push(stackIn.pop)
String toDequeue = stackOut.pop
4. Dequeue for list: $O(1)$ time complexity $O(1)$ space complexity
Dequeue for array: $O(1)$ time complexity $O(1)$ space complexity
Dequeue for stack: $O(n)$ time complexity $O(1)$ space complexity
5. Does not apply.
6. I liked implementing the circular array. I had previously never knew about this concept before and it's interesting to think about that you need to different "sizes" to always consider, the size of the entire "physical" array and the size of the array you actually want. I think the specifications were a little bit unclear. It was confusing what exactly the program was supposed to do and there wasn't much of an example output. I personally had trouble with dealing with the generic array case. I had three pairs of nearly identical methods, except that one use an ArrayQueue and the other a ListQueue.