Lab 03 – Arrays and Linked List Queues

# Problem

We were issued the task of designing multiple Java classes. We had to create two queues, one utilizing an Array and one utilizing a Linked List. Both needed to be generic types, and have several constraints including certain constructors, methods, and instance variables. The linked list needed to have an internal class ListNode, in addition to some special pointers. Both the array and linked list needed to be equipped to handle enqueing and dequeing, and be able to perform the tasks designated in the driver class.

# Proposed Solution

Based on the example demonstrated in class, I was easily able to get a working version of the GenArrayQueue. The GenLLQueue was a little harder, but I feel much better about linked lists after working through this lab. Implementing both the array and linked list variants of the queues were very simple.

# Tests and Results

I had successful tests with the driver class after implementing the two types of queues. The results were as displayed in the given example.

# Problems Encountered

I actually wasted a solid 1.5 hours trying to code my own driver class, and having issues when attempting to typecast variables to the data retrieved from the .data portion of the linked list. It wasn’t until after getting truly stuck on this that I noticed a provided driver class. I updated my method names to match the provided driver class, and it ran perfectly on the first try. I will make sure to fully read the lab example and directions next time, as this was a very time costly mistake on my part.

# Conclusions and Discussions

I feel that I have a much stronger grasp of linked lists after this lab, and have further strengthened my skillset in regards to arrays.

# Additional Questions

1. Given this array representation of a Queue with the following operations, what will the queue look like at the end?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| VALUE | 22 | 4 | 8 | 2 |  |  |  |  |  |

Operations:

1. Enqueue 51, 50 ,33

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| VALUE | 22 | 4 | 8 | 2 | 51 | 50 | 33 |  |  |

1. Dequeue three times

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| VALUE | 2 | 51 | 50 | 33 |  |  |  |  |  |

1. Enqueue 34, 2, 6, 7

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| VALUE | 2 | 51 | 50 | 33 | 34 | 2 | 6 | 7 |  |

1. Dequeue twice

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| VALUE | 50 | 33 | 34 | 2 | 6 | 7 |  |  |  |

1. Repeat question 1 but redraw it as a singly linked list showing each step.

8

4

22

33

50

51

2