

## Programming Assignment 3 Part II

### Carson Hanel

Preface: Alright, I realize this isn't LyX but I thought this would be nicer to look at than a shoddily put together LyX generated PDF. Not much in the way of guidelines was provided, so I'm sort of winging this, but trying to hit the major points. I'll probably copy and paste this to the head of the other page as well. It is highly helpful to pair this document with my code comments.

1. Assignment description:
  - a. The assignment was to implement a MinQueue based upon a Doubly Linked List. The MinQueue has elements size which is an integer holding the size of the queue, and line which is the internal Doubly Linked List. The purpose of this MinQueue was apparently supposed to function exactly like a regular queue, but with the ability to find the minimum element. I personally thought this to be incredibly counter-intuitive especially with respect to how MinQueue was used in lecture, but when I asked and looked at the directions, this is what we had. It was a fun assignment, I had a little frustration getting it to run on Unix, but in the end it was a fun time.
2. Description of DS&A used:
  - a. The data structures used include:
    - i. The class Node
    - ii. The class DoublyLinkedList
    - iii. The class MinQueue
  - b. The algorithms used include:
    - i. Linear search using pointers when searching for min
    - ii. Constant removal from end
    - iii. Constant insertion at beginning
  - c. How I solved the problems:
    - i. The main functions of MinQueue:
      1. Enqueue  $O(1)$ : Adds an object of type T to the back of the queue.
      2. Dequeue  $O(1)$ : Removes an object of type T from the front of the queue.
      3. Min  $O(n)$ : Linearly searches the queue for the current lowest value, and returns the value of that node's object.
      4. isEmpty  $O(1)$ : Returns true if the variable size is zero.
3. A user guide to navigate my program:
  - a. Access the MinQueue folder
  - b. Type make into the environment, I made my own makefile for it.
  - c. Type ./a.out or ./QueueMain
4. Types of exceptions:
  - a. Dequeueing from an empty list
  - b. Min from an empty list
5. Testing:

a.

```
[chingyl510]@compute ~/221/P3/TemplateDoublyLinkedList/MyQueue> (23:14:23 07/29/1
:: ./QueueMain
Enqueue of: 1
Enqueue of: 9
Enqueue of: 2
Enqueue of: 8
Enqueue of: 3
Enqueue of: 7
Enqueue of: 4
Enqueue of: 6
Enqueue of: 5
Enqueue of: 5
Enqueue of: 6
Enqueue of: 4
Enqueue of: 7
Enqueue of: 3
Enqueue of: 8
Enqueue of: 2
Enqueue of: 9
Enqueue of: 1
1 was the lowest value in the list.
Dequeue of: 1
1 was the lowest value in the list.
Dequeue of: 9
1 was the lowest value in the list.
Dequeue of: 2
1 was the lowest value in the list.
Dequeue of: 8
1 was the lowest value in the list.
Dequeue of: 3
1 was the lowest value in the list.
Dequeue of: 7
1 was the lowest value in the list.
Dequeue of: 4
1 was the lowest value in the list.
Dequeue of: 6
1 was the lowest value in the list.
Dequeue of: 5
1 was the lowest value in the list.
Dequeue of: 5
1 was the lowest value in the list.
Dequeue of: 6
1 was the lowest value in the list.
Dequeue of: 4
1 was the lowest value in the list.
Dequeue of: 7
1 was the lowest value in the list.
Dequeue of: 3
1 was the lowest value in the list.
Dequeue of: 8
1 was the lowest value in the list.
Dequeue of: 2
1 was the lowest value in the list.
Dequeue of: 9
1 was the lowest value in the list.
Dequeue of: 1
No minimum in an empty queue!
Cannot dequeue from an empty queue!
```