Chris W Jones

CMSC350 – 7380

5 July 2015

# Homework 2 Documentation

## Analysis

I started the homework by reading over the requirements given in class to make sure that there wasn’t anything too wild. Since this week’s assignment was a linked list, I wouldn’t be reinventing the wheel. I read through the section on linked lists in Cormen’s Introduction to Algorithms (2001). That section gave me an overview of the algorithms that I’d need to implement and an outline of how to do it in pseudocode.

## Design

I feel as though most of the design decision on this project were out of my hands. Since I was essentially retreading old ground, I had plenty of examples of how to design a linked list implementation. There was only one aspect that I didn’t see many places, a sentinel.

I used a special node with a value of null to keep track of the first and last nodes in the list. This allowed me to eliminate checking for null in my code since the linked list was circular with this sentinel object.

## User Interface

I purposefully included no user interface. While there are many situations that could be modeled by a GUI and represented by a linked list, I didn’t believe that this homework needed it. It would have increased the codebase and time required quite a bit. Also, since the purpose of this assignment is to show our understanding of a linked list and not our GUI writing ability. It was for those reasons that I decided to simply output all of the details to System.out.

## Testing

I did a good bit of testing of this homework, mainly because I didn’t think through a lot of my methods before starting to write them. I found several times where I was incorrectly setting or reading the next or previous Node in the list. It was so bad that I ended up using break points and stepping through the insert function while watching the composition of the list so that I could get it correct.

## Error Handling

There were several spots in the code where I needed to implement error handling. Since there are several ways to pass incorrect data, mainly in the delete() and search() methods. Since in those methods it is possible to pass a node that is not in the list, I wrote a new exception for when a node is not in the list. I also implemented a try/catch block that shows this error being raised.

## Conclusions

A linked list is a fundamental data structure. Doing this homework has given me a much deeper understanding of how they work, especially in a language that doesn’t have (easy) direct memory access.

References

Cormen, T. (2001). *Introduction to algorithms* (2nd ed.). Cambridge, Mass.: MIT Press.