Koby Miller

Project 1

Project One Redo: Adding and Multiplying Matrices

To start off, I used a lot of the same code from the first project, so I will briefly explain how I did that. I coded that project using a linked list. I created a Node class that was used in each link of the linked list. Each Node had an int for its data and 2 other ints to keep track of that nodes row and column in the matrix. Lastly, each node would also have a next and prev nodes. When a piece of data is added a new node is made and put in the list. If the data is 0, there is no node (so it won’t take up any memory).

I chose this method because honestly it was the first thing that came to mind. I knew I needed to find a way to not allocate memory for 0’s. So, I thought of how I could create a coordinate system and if the specific coordinate was empty than it would really be 0. With one list I thought I would be able to reach any information needed easily.

Moreover, the adding method that I created is very simple. All it does is traverse one list and add any numbers it does have to the same coordinates in the second matrix. First, I check to see if the sizes of the matrices are the same (even though you can add two different sizes, but the project wanted us to check this. From there I just used a while loop and when through the first matrix.

Furthermore, I created a small method called dotProduct to help me with the multiplying. I could have put it all in the same method, but I could think of it better separated. To keep it short, I used two loops to find the answer for a specific coordinate, one loop to traverse the new matrix and place the answer found. The first two loops were to go through each of the matrices being multiplied and find the dot product out of those rows/columns.

I had a couple issues at the end and could not figure them out and then while trying to fix my code I confused myself and started thinking that certain variables were keeping track of different things. For example, NumCol makes me think it is counting the columns but really the way it was used to keep track of what column you were on. So, I realized I had confused variables throughout the whole code. This didn’t make much difference in the tested code because most of the matrices were squared meaning the size didn’t have an effect. I believe I fixed most of the variable.

Computational complexity is in the comments on the code.