My sparse matrix was implemented using a class called SparseMatrix that had three pieces of data. The first was a Linked List that held all the nonzero elements contained in the sparse matrix. The other two were integers containing the number of rows and the number of columns in the matrix. The elements that were contained within the Linked List were of a class that I created called “element.” This class contained each particular element’s row and column in the sparse matrix and its corresponding data value. Elements were only created if they were nonzero.

I chose this particular implementation because it was the most straightforward and easy to see, and to me, the easiest to implement. This implementation also made it so that elements with data equal to 0 did not have to be allocated in memory. The computational complexity of the methods that are contained within the SparseMatrix class are as follows:

- clear – O(1)
- setSize – O(1)
- getNumRows – O(1)
- getNumCols – O(1)
- addElement – O(n)
- removeElement – O(n)
- getElement – O(n)
- toString – O(n)
- addMatrices – O(n^2)
- multiplyMatrices – O(n^3)