Turn in a one-paragraph commentary with your code. Describe your matrix implementation. Why did you choose the implementation you did? What is the computational complexity of the operations in your matrix implementation?

My matrix implementation consists of using Java's built-in LinkedList class with my own "matrixNode" object to serve as the nodes of this linked list. The matrixNode stores the row, column, and value of non-zero values in the matrix. I also stored the size of the matrix to make sure that values outsize of the matrix bounds could not be added. Because the initial element of the matrix is located at (0,0) rather than (1,1), I must compare values to the difference between the size and one rather than the stored size value. I chose this way of implementation because this was the simplest, most logical way for me to meet the project specifications. The computational complexity of the operations in my matrix implementation are as follows:

- clear – O(1)
- setSize – O(1)
- addElement – O(n)
- removeElement – O(n)
- getElement – O(n)
- determinant – O(n!)
- minor – O(n)
- toString – O(n)
- getSize – O(1)