Marcus Mills

COP3530

Project 1

Recursive Determinant Commentary

My SparseMatrix class uses a custom made linked list to store the non-zero elements in a matrix. It does this by storing the row, column, and data of each non-zero element as nodes. The reason why I chose this way of organizing my data is because you only need to store the data of non-zeros, as opposed to storing everything. As for the reason why I chose a linked list to store my data, as opposed to an array, is because I can add and remove elements without having to create a new list every time. This process of adding and removing elements takes O(n) time. The reason why it’s not O(1) time is because my SparseMatrix class stores nodes in order based firstly on row, then column. As for the other functions, toString(), getElement(), and minor() also take O(n) time, while clear(), setSize(), and getSize() take O(1) time. The only oddball out is determinant(). This function takes O(n!) time, since it is iterative, recursive, and calls minor each iteration.