Gaussian elimination:

a. The outer loop iterates over each row of the matrix (i represents the current row).

b. Within each iteration, the algorithm finds the pivot row. It starts from the current row (i) and searches for a row with a larger absolute value at the current column (j = i).

c. If a row with a larger absolute value is found (abs(matrix.data[j][i]) > abs(matrix.data[pivotRow][i])), the pivot row index is updated (pivotRow = j).

d. After finding the pivot row, the current row and the pivot row are swapped if they are different. This ensures that the pivot element (element at (i, i)) has the maximum absolute value among all elements in the column below it.

e. If the pivot element is zero (matrix.data[i][i] == 0.0), it means the matrix is singular (no inverse exists), and a runtime error is thrown.

f. The current row is then scaled so that the pivot element becomes 1.0. This is done by dividing the entire row by the value of the pivot element (scale = matrix.data[i][i]).

g. Next, the algorithm eliminates the elements below the pivot in the current column. It iterates over each row below the current row (j = i + 1) and subtracts a scaled multiple of the pivot row from it. This ensures that all elements below the pivot become zero.

Back substitution:

a. The algorithm performs back substitution to obtain the inverse matrix. It starts from the last row (i = rows - 1) and iterates backwards up to the second row (i > 0).

b. Within each iteration, it iterates over each row above the current row (j = i - 1) and subtracts a scaled multiple of the current row from it. This ensures that all elements above the diagonal become zero.

The resulting matrix (identity) contains the inverse of the original matrix.

In summary, the algorithm uses Gaussian elimination to transform the original matrix into an upper triangular form and then performs back substitution to obtain the inverse matrix. It handles singular matrices by throwing a runtime error. The algorithm utilizes row operations to manipulate the matrix and identity matrix simultaneously, ensuring the resulting identity matrix holds the inverse of the original matrix.