HW DSA Complexity

Task 1

The expected running time of the code is ~ O(n*n). The outer loop performs n iterations, and the inner while-loop (which is nested) performs less than n, but since - roughly n iterations. The two loops are nested, so the complexity is n*n.

Task 2

The outer for-loop performs n iterations and the nested loop performs m iterations at most, depending on whether the elements of the first column of the matrix are even, so the running time of the code is ~O(n*m). At best, the whole first column is odd and then the complexity is just n.

Task 3*

To start with, GetLength(0) and GetLength(1) are put the wrong way and need to be swapped, otherwise IndexOutOfRangeException would be thrown if rows > columns.

Now having the code fixed, the number of recursive calls is roughly equal to the number of rows n, since the initial call of the function starts from parameter row = 0 and “row + 1 < matrix.GetLength(0)” is just equal to “if row < n”.

For each recursive call a for-loop is performed which takes exactly m iterations (for each column from 0 to m). Thus the code complexity is O(m*n).

If we keep the code as it is and assume that rows <= columns, the for-loop performs n iterations and the recursive call performs n – 1 iterations and then throws the exception mentioned above. In this case the complexity is O(n*n).