Each row is scaled roughly one time so there are roughly n scaling operations, each of which requires n operations. The number of operations due to scaling is roughly n^2.

Therefore, the total number of operations is roughly

$$2n^3 + n^2.$$

When n is very large, the n^2 term is much smaller than the n^3 term. We therefore state that

Observation 1.3.2 The number of operations required to find the reduced row echelon form of an $n \times n$ matrix is roughly proportional to n^3.

This is a very rough measure of the effort required to find the reduced row echelon form; a more careful accounting shows that the number of arithmetic operations is roughly $\frac{2}{3}n^3$. As we have seen, some matrices require more effort than others, but the upshot of this observation is that the effort is proportional to n^3. We can think of this in the following way: If the size of the matrix grows by a factor of 10, then the effort required grows by a factor of $10^3 = 1000$.

While today's computers are powerful, they cannot handle every problem we might ask of them. Eventually, we would like to be able to consider matrices that have $n = 10^{12}$ (a trillion) rows and columns. In very broad terms, the effort required to find the reduced row echelon matrix will require roughly $(10^{12})^3 = 10^{36}$ operations.

To put this into context, imagine we need to solve a linear system with a trillion equations and a trillion variables and that we have a computer that can perform a trillion, 10^{12}, operations every second. Finding the reduced row echelon form would take about 10^{16} years. At this time, the universe is estimated to be approximately 10^{10} years old. If we started the calculation when the universe was born, we'd be about one-millionth of the way through.

This may seem like an absurd situation, but we'll see in Subsection 4.5.3 how we use the results of such a computation every day. Clearly, we will need some better tools to deal with *really* big problems like this one.

1.3.4 Summary

We learned some basic features of Sage with an emphasis on finding the reduced row echelon form of a matrix.

- Sage can perform basic arithmetic using standard operators. Sage can also save results from one command to be reused in a later command.
- We may define matrices in Sage and find the reduced row echelon form using the `rref` command.
- We saw an example of the Augmentation Principle, which we then stated as a general principle.
- We saw that the computational effort required to find the reduced row echelon form of an $n \times n$ matrix is proportional to n^3.

Appendix A contains a reference outlining the Sage commands that we have encountered.