Project Submission Writeup for Efficient Matrix Data Structure

How the project works:

I implemented two data structures: DenseMatrix and SparseMatrix which carries out the usual matrix operations such as addition, multiplication, transpose, etc. However, for this project, their operations are carried out using the idea of parallelism. This would allow for better speedups and performance for those who would like to use it.

In order to run the methods first, you will have to instantiate the classes so for DenseMatrix all you will need to provide is a 2D Array of ArrayBuffer while for SparseMatrix since it takes advantage of DOK (keys are coordinates in tuple format of (row, col) and the value is the non zero elements) representation it will need to take in a Map, worry not Matrix Object has a method written to convert a 2D array of ArrayBuffer to a Sparse Matrix.

After that simply call any of the methods (note for addition, multiplication, equal, etc. another corresponding matrix type is needed obviously)

Findings:

Even though the runtime of the parallel method varied for each run, the overall average runtime was faster than the sequential version for the most part thanks to parallel collections by scala.

Things I learned:

The first thing I found was there are parallel collections in scala and I was shocked at how easy it is to use straight away after adding the dependency.

It also helped improve the runtime of my code when compared to the sequential version. Before the project, I also never heard about sparse matrices and dense matrices so it was very fun and enjoyable reading about their differences and how they work especially for sparse matrices as they provide a lot of use in saving memory and machine learning.

Lastly, I got the chance to learn more about linear algebra such as I didn’t know that we could find determinants by first decomposing the matrix and then finding the product of diagonals to get the determinant.

Anything extra to tell:

The project still is not perfect and definitely need improvement such as the determinant of sparse matrices sometimes doesn't give the correct output as I think because the Gaussian elimination doesn’t work well with many 0s element in a matrix essentially sparse matrice

The project may still be improved and worked on later on even after the submission deadline as I really enjoyed learning about matrices and coding on scala just feels amazing (definitely one of my top 3 languages of all time).

I hope someone in this field may find my project useful and use it for their own projects.