# **Report -** Empirical analysis of algorithms based on scientific principles.

Empirical Analysis of Matrix Multiplication Algorithms:

The purpose of the homework assignment is to learn how to do empirical analysis of algorithms based on scientific principles. The traditional matrix multiplication algorithm performs very well for small matrices and does poorly on problems of large sizes. On the other-hand, Strassen’s matrix multiplication algorithm is performs efficiently on problem sizes that are large. You are combine the two algorithms and general and study if there is a way to design a new algorithm that leverages the strength of both while reducing their negative side. In your empirical examination, you must determine the optimum break-even point. We need to see good data gathering, evidence of fact, method used, how you dealt with the flacutations in measurements, graphs, predictions, etc. Also, studying other algorithms with even more efficiencies than Strassen’s. For extra points, If implement all of the above in the C programming language, compare the performance of the C implementation to that of Java.

Criteria for assignment 3:

1 Correct method and answer to calculate matrix multiplication with matrix reading from given file using traditional algorithm -10’

2 Correct method and answer to calculate matrix multiplication with matrix reading from given file using Strassen’s matrix multiplication -10’

3 Write code to get break point and provide data you will use to support your conclusion in report \*automatically\* (which means you will not enter the size of matrix manually). The input could be null, output will be break point and some other data you will use in report. (code - 20’, reasonable break point answer - 10’)

4 Improved algorithm to calculate matrix multiplication considering the size of input matrix - 30’

5 Well-structured report with visualized data which supported your conclusion - 20’ (Here is just an example of report structure: \*Introduction\* – briefly describe this problem, \*Method\* – your method or key part of code to implement traditional matrix multiplication and Strassen’s matrix multiplication, and get the break point, \*Results\* – the data you get which will be used to help further analyze and support your conclusion, \*Discussion\* – analyze about break point and the reason you decide it as the break point, \*Conclusion\* – the conclusion you get from the analyze, the describe and explanation of the improved matrix multiplication algorithm)

6 Comparison between the performance of C implementation and Java implementation, which is included in proper section of the report. – 20’

Your report should also contain screenshot and short explanation for output of criterion 1, criterion 2, criterion 4 and break point. Otherwise you won’t get points of that criteria.

For criterion 4: You need to demonstrate the advantage of the improved algorithm comparing with traditional way and Strassen’s matrix multiplication with evidence, eg: calculate the matrix multiplication with same matrix in three algorithms, and provide the screenshot of running time (the size of the input matrix must greater than breakpoint, and you can create the matrix with random numbers).

For criterion 6: To get extra point, you must write all of the code in both Java and C, and include both screenshot in report, and discuss the running time in report.

Be aware that the size of input matrix must be power of 2. Consider that when increasing the matrix size in criterion 3.

or assignment 3, if you cannot calculate the multiplication of matrix which size over 500 with Strassen’s algorithm due to the long running time, you can consider combining criterion 3&4. It means you can make several assumption of break point with improved algorithm, do the same comparison among traditional algorithm, Strassen’s algorithm and those different assumptions. The result can be shown in same diagram or table in your report. But your report should also contains the running time of Strassen’s algorithm and simple explanation of the result you get if the running time is too long to accept.

One of the reason that Strassen’s algorithm takes long time in small size matrix is that it needs much more memory space to store the result of sub-matrix multiplication.

## Introduction

Empirical analysis of algorithms based on scientific principles:

The traditional matrix multiplication algorithm performs very well for small matrices and does poorly on problems of large sizes. On the other-hand, Strassen’s matrix multiplication algorithm is performs efficiently on problem sizes that are large.

The aim is to combine the two algorithms in general and study if there is a way to design a new algorithm that leverages the strength of both while reducing their negative side by determining the optimum break-even point.

## Method

Your method or key part of code to implement traditional matrix multiplication and Strassen’s matrix multiplication, and get the break point,

## Results

The data you get which will be used to help further analyze and support your conclusion,

## Discussion

Breakpoint: The point where the time complexity of the traditional matrix multiplication becomes more than the time complexity of the Strassen’s algorithm.

Logic to find the breakpoint:

The input size of the matrix is set to 2^n where n starts with a value of 1 and increases by 1 with every loop till the time complexity of Strassen’s algorithm is less than the time complexity of Traditional algorithm. The time complexity is calculated for each increment on n for both the algorithms.

As soon as the time take by Traditional method goes beyond the time taken by Strassen’s algorithm, the loop breaks and the size of the matrix at that point is the break-even point.

## Conclusion

The conclusion you get from the analyze, the describe and explanation of the improved matrix multiplication algorithm