

## Algorithms Design and Analysis (CSCI-B503)

### Homework #3 Solutions – Gourav Shenoy

Q: Exercise 6.5-9 and Problem 6-2

A: Handwritten paper copy of the answers have been submitted to the professor on Feb 08, 2016 in class. I will also be submitting a soft copy (pdf) of the answer sheet on Canvas.

Q: Matrix Multiplication Program.

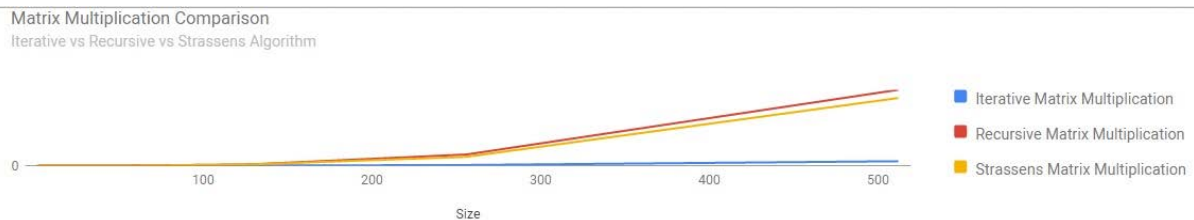
A: I will be submitting the program (python code – got approval from professor for python programming) on Canvas. I will also be attaching a README.md file giving instructions.

Along with the README.md file, I will also be attaching the output file (matrix\_mult.out) that I got on sample run of my program. I will also be turning in my graph plot (see below).

Q: time and plot the results of standard vs Strassen's matrix multiply.

A: I have compared the running time of the 3 matrix multiplication techniques (Iterative, Recursive, and Strassen's) and plotted the results as a graph. Please find the output image file (Matrix\_Mult\_Comparison.jpg) on Canvas submission.

Comparisons are made for input sizes: 2x2, 4x4, 8x8, 16x16, 32x32, 64x64, 128x128, 256x256, and 512x512.



Q: Try to find what size matrix will cause Strassen's to equal standard matrix multiply. It should be somewhere between 40 and 500 depending on your implementation.

A: I noticed that recursive matrix multiplication & Strassen's matrix multiplication run times are similar for sizes up to 128x128. After this, Strassen's outperforms recursive approach.

**Note:** This is as per my implementation.

Q: write up and discuss your results.

According to my implementation:

- Strassen's matrix multiplication is faster for larger input sizes (>128x128).
- Recursive method and Strassen's method run times are almost similar for input sizes <128x128.
- Strassen's method is extremely inefficient for smaller input sizes.

Q: Could Strassen's be improved for smaller matrices? What are the inefficiencies of Strassen's, could this be improved?

A: Strassen's method is extremely inefficient for small input sizes.

- It consumes a lot of memory and also takes time.
- It is not worth the amount of computational resources, since a simple iterative multiplication will run faster for small matrices.
- One approach to improve Strassen's would be to make it a parallel algorithm – E.g.: Using threads to run parallel and thereby running faster.
- In this parallel approach, after the division of matrices into submatrices, separate threads can be used to perform the 7 recursive sub-matrix multiplications.
- Also, for smaller input sizes we could perform a simple naïve algorithm instead of having 7 multiplications via divide and conquer.