

Tofik Mussa

Programming Assignment 2 Writeup

There were 3 tasks to be performed in this assignment: to analytically determine a crossover point beyond which Strassen's algorithm doesn't yield any advantages to running the traditional matrix multiplication algorithm, to experimentally prove the same, and to compute the triangle in random graphs representing an adjacency matrix. Implementation of my algorithm was inspired by some excerpts the Algorithm Design book by authors Kleinberg and Tardos as well as CLRS.

Analytically determining crossover point

The recurrence for Strassen's algorithm is $T(n) = 7T(n/2) + 18*((n/2)^2)$. There are 7 subproblems of size $(n/2)$ and 18 primitive operations on matrices of size $(n/2)$ costing a total of $18*((n/2)^2)$.

The recurrence for traditional matrix multiplication is $T'(n) = (n^2) * (2n - 1)$ because there is (n^2) elements and there are $(2n - 1)$ to compute each cell.

Doing some algebra with equating these 2 equations and plugging in Desmos, we find that they intersect at 15/16.

Experimentally determining crossover point

I run and collected datapoints to determine crossover points in both odd and even number of dimensions. For the even case, the gain in saved running time tapered off for cross over point values greater than 16. There were no significant gains in time even after increasing the crossover point to 80 and 160. This was in line with the analytical derivations as well. Changing the crossover point from 15 to 16 drastically cut the runtime by more than half but increasing it to more than 16 didn't yield much improvement.

Random Triangle Graph

I implemented a random graph with random numbers generated from 0 to 2 as the entries of the adjacency matrix with the following probabilities of getting included. The numbers are pretty close to what is provided by the formula in the requirements.

P | dimension | predictedValue | expectedValue

0.01 | 1024 | 184.16666666666666 | 178.4

0.02 | 1024 | 1435.3333333333333 | 1427.46

0.03 | 1024 | 4745.0 | 4817.69

0.04 | 1024 | 11500.5 | 11419.71

0.05 | 1024 | 22515.333333333332 | 22304.13

Implementation Details

I implemented a general algorithm for non-square matrices. I padded one column of zeroes and one row of zeroes when an odd dimension is provided. The result is copied omitting the zeroes after the computation is performed. I had some issues wrapping my head around the recursion involved in Strassen's algorithm and working out the product of matrices by hand for small inputs helped. One off error due to miscalculating rows and columns were also things I was on the lookout for. Determining the crossover point was also done by running a series of experiments and determining where the runtime stops improving.

I chose Java for the implementation because of my familiarity with it and because there is no burden of manual memory management. Despite having run into heap space issues in the previous assignment, I didn't have those issues in this assignment. I have tested with 2048X2048 matrices and they were able to be held in memory with no issues. I also tested with odd and even dimensions and it worked as expected.

I have used 7 multiplications instead of 8 using Strassen's and the constant time operations like addition, subtraction... didn't consume much time which experimentally validated the initial assumption.

Data Points used to determine crossover point – Even Case

crossOverPoint | dimension | timeTaken

4 | 4 | 0
4 | 8 | 0
4 | 16 | 0
4 | 32 | 0
4 | 64 | 0
4 | 128 | 0
4 | 256 | 0
4 | 512 | 1
4 | 1024 | 6
4 | 2048 | 47

Data 1.1 - Runtime in seconds setting 4 as crossover point for values starting at 4

crossOverPoint | dimension | timeTaken

8 | 4 | 0
8 | 8 | 0
8 | 16 | 0

8		32		0
8		64		0
8		128		0
8		256		0
8		512		0
8		1024		2
8		2048		15

Data 1.2 - Runtime in seconds setting 8 as crossover point for values starting at 4

crossoverPoint | dimension | timeTaken

12		4		0
12		8		0
12		16		0
12		32		0
12		64		0
12		128		0
12		256		0
12		512		0
12		1024		2
12		2048		15

Data 1.3 - Runtime in seconds setting 12 as crossover point for values starting at 4

crossoverPoint | dimension | timeTaken

16		4		0
16		8		0
16		16		0
16		32		0
16		64		0
16		128		0
16		256		0
16		512		0

16 | 1024 | 1

16 | 2048 | 7

Data 1.4 - Runtime in seconds setting 16 as crossover point for values starting at 4

crossoverPoint | dimension | timeTaken

20 | 4 | 0

20 | 8 | 0

20 | 16 | 0

20 | 32 | 0

20 | 64 | 0

20 | 128 | 0

20 | 256 | 0

20 | 512 | 0

20 | 1024 | 1

20 | 2048 | 7

Data 1.5 - Runtime in seconds setting 20 as crossover point for values starting at 4

crossoverPoint | dimension | timeTaken

24 | 4 | 0

24 | 8 | 0

24 | 16 | 0

24 | 32 | 0

24 | 64 | 0

24 | 128 | 0

24 | 256 | 0

24 | 512 | 0

24 | 1024 | 1

24 | 2048 | 7

Data 1.6 - Runtime in seconds setting 24 as crossover point for values starting at 4

crossoverPoint | dimension | timeTaken

28 | 4 | 0

28		8		0
28		16		0
28		32		0
28		64		0
28		128		0
28		256		0
28		512		0
28		1024		1
28		2048		7

Data 1.7 - Runtime in seconds setting 28 as crossover point for values starting at 4

crossOverPoint | dimension | timeTaken

32		4		0
32		8		0
32		16		0
32		32		0
32		64		0
32		128		0
32		256		0
32		512		0
32		1024		0
32		2048		5

Data 1.8 - Runtime in seconds setting 32 as crossover point for values starting at 4

crossOverPoint | dimension | timeTaken

36		4		0
36		8		0
36		16		0
36		32		0
36		64		0
36		128		0

36		256		0
36		512		0
36		1024		0
36		2048		5

Data 1.9 - Runtime in seconds setting 36 as crossover point for values starting at 4

crossoverPoint	 	dimension	 	timeTaken
-----------------------	----------	------------------	----------	------------------

40		4		0
40		8		0
40		16		0
40		32		0
40		64		0
40		128		0
40		256		0
40		512		0
40		1024		0
40		2048		5

Data 1.10 - Runtime in seconds setting 40 as crossover point for values starting at 4

crossoverPoint	 	dimension	 	timeTaken
-----------------------	----------	------------------	----------	------------------

80		4		0
80		8		0
80		16		0
80		32		0
80		64		0
80		128		0
80		256		0
80		512		0
80		1024		0
80		2048		4

Data 1.11 - Runtime in seconds setting 80 as crossover point for values starting at 4

crossOverPoint | dimension | timeTaken

160 | 4 | 0
160 | 8 | 0
160 | 16 | 0
160 | 32 | 0
160 | 64 | 0
160 | 128 | 0
160 | 256 | 0
160 | 512 | 0
160 | 1024 | 0
160 | 2048 | 4

Data 1.12 - Runtime in seconds setting 160 as crossover point for values starting at 4

Data Points used to determine crossover point – Odd Case

crossOverPoint | dimension | timeTaken

4 | 3 | 0
4 | 7 | 0
4 | 15 | 0
4 | 31 | 0
4 | 63 | 0
4 | 127 | 0
4 | 255 | 0
4 | 511 | 1
4 | 1023 | 7
4 | 2047 | 49

Data 2.1 - Runtime in seconds setting 4 as crossover point for odd values starting at 3

crossOverPoint | dimension | timeTaken

8 | 3 | 0
8 | 7 | 0
8 | 15 | 0

8		31		0
8		63		0
8		127		0
8		255		0
8		511		0
8		1023		2
8		2047		15

Data 2.2 - Runtime in seconds setting 8 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

12		3		0
12		7		0
12		15		0
12		31		0
12		63		0
12		127		0
12		255		0
12		511		0
12		1023		2
12		2047		15

Data 2.3 - Runtime in seconds setting 12 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

15		3		0
15		7		0
15		15		0
15		31		0
15		63		0
15		127		0
15		255		0
15		511		0

15 | 1023 | 2

15 | 2047 | 15

Data 2.33 - Runtime in seconds setting 15 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

16 | 3 | 0

16 | 7 | 0

16 | 15 | 0

16 | 31 | 0

16 | 63 | 0

16 | 127 | 0

16 | 255 | 0

16 | 511 | 0

16 | 1023 | 1

16 | 2047 | 7

Data 2.4 - Runtime in seconds setting 16 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

17 | 3 | 0

17 | 7 | 0

17 | 15 | 0

17 | 31 | 0

17 | 63 | 0

17 | 127 | 0

17 | 255 | 0

17 | 511 | 0

17 | 1023 | 1

17 | 2047 | 7

Data 2.44 - Runtime in seconds setting 17 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

20 | 3 | 0

20		7		0
20		15		0
20		31		0
20		63		0
20		127		0
20		255		0
20		511		0
20		1023		1
20		2047		7

Data 2.5 - Runtime in seconds setting 20 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

24		3		0
24		7		0
24		15		0
24		31		0
24		63		0
24		127		0
24		255		0
24		511		0
24		1023		1
24		2047		7

Data 2.6 - Runtime in seconds setting 24 as crossover point for odd values starting at 3

crossoverPoint | dimension | timeTaken

28		3		0
28		7		0
28		15		0
28		31		0
28		63		0
28		127		0

28 | 255 | 0

28 | 511 | 0

28 | 1023 | 1

28 | 2047 | 7

Data 2.7 - Runtime in seconds setting 28 as crossover point for odd values starting at 3