Chris Pardee
Austin Williams
CS 124
3/22/21

Benchmarking Strassen's Algorithm

For this assignment, we were tasked with assessing what would be the optimal crossover point for Strassen's algorithm. We asked "at what point do we stop the recursion and use the naive multiplication algorithm for the base case?". For part 1 of the assignment, we had to calculate the crossover point using the recurrence. The recurrence for Strassen's algorithm is $T(n) = 7T(\frac{n}{2}) + 18n^2$. This is because there are 7 different products that we have to calculate recursively, and there are 18 addition or subtraction steps where we add or subtract matrices element-wise (which is n^2 time). In order to find the crossover point, we consider the inequality $7T(\frac{n}{2}) + 18n^2 < n^3$, which compares strassen's algorithm to the runtime of the naive multiplication algorithm. We do this comparison because we want to know at what point is strassen's algorithm faster than the naive algorithm. First we use the master theorem to solve the recurrence. Using the master theorem, the recurrence simplifies to $T(n) = n^{log_2 7} + 18n^2$. So we can solve the inequality $n^{log_2 7} + 18n^2 < n^3$and we get that the estimated crossover point is 36. This gets us the best crossover point in theory.

For part 2 of the assignment, we implemented Strassen's algorithm as well as the naive multiplication algorithm and ran tests to discover the optimal crossover point in practice. We tested matrix sizes 512, 1024, 1536, 2048, and 4096 on crossover points 32, 64, 128, 256, 512, and 1024. Since the multiplications run in roughly n^3 time, we weren't able to test all of the crossover points for matrices of size 4096, simply because the trials took many hours to complete. We did include data for all the crossover points we were able to test for size 4096 matrices. At the end of this document are tables and graphs describing the data we collected from these tests.

What we discovered in the data is that 64 as the crossover point minimized the running time of the matrix multiplications for most of the matrix sizes. The only cases where 64 wasn't the minimum running time was for size 1024 and size 2048 matrices. In the case of 1024 size matrices, the crossover point 64 is roughly tied with 32 and 128. All three crossover points yielded a running time within a second of each other. So for this case 64 is effectively tied for the best running time because of this negligible difference. For 2048 size matrices, the crossover point 256 does better. In this case, 64 yields a time of 1386 seconds and 256 yields a time of 969 seconds. However for size 1536 matrices, the crossover point 256 yields the max runtime by a wide margin. So this disqualifies 256 from our consideration of the best crossover point. Overall, 64 is the most consistent minimum or near minimum running time for Strassen's algorithm that we identified.

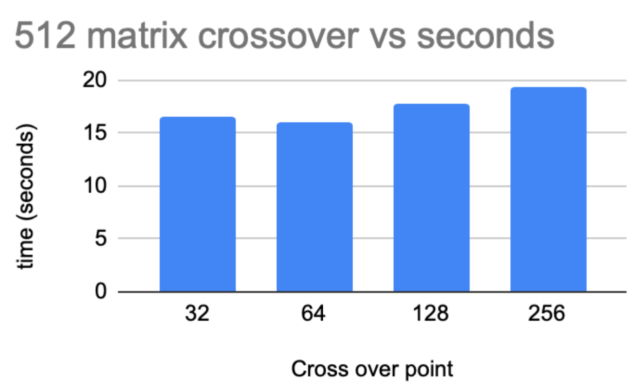
Our estimation for the best crossover point in part 1 was 36. The crossover point 32 is the closest matrix size we tested, since it is the closest power of 2. Strassen's algorithm divides a matrix into four matrices that have dimensions that are a power of 2, so 36 cannot be used as a crossover point as in our estimation. 32 as the crossover point did not perform as well at reducing the running time of Strassen's algorithm as 64 did. This is evidenced in the data included at the end of this document. So our prediction of the best crossover point did not actually result in the best crossover point in practice.

For part 3 of the assignment, we were tasked with calculating the number of triangles in size 1024 random graphs. The random graphs are constructed where each edge is included with a certain probability p. The probability p can be one of [0.01, 0.02, 0.03, 0.04, 0.05]. As the probability p increases, the number of edges in the random graph increases. This is because the expected number of edges increases with the probability. We found that the number of triangles in the graph increased with the number of edges included. As the probability increases, the number of triangles increases as well. At the end of this document is a chart demonstrating the number of triangles generated for each probability of edge inclusion.

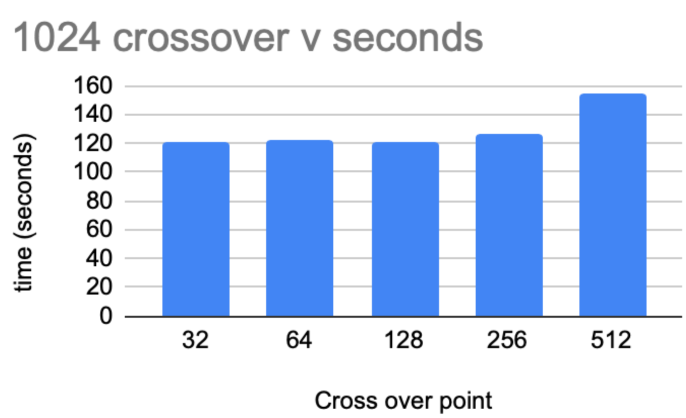
As shown in the data, the number of triangles generated from Strassen's algorithm is very close to the expected number of triangles resulting from the expression $(1024\ choose\ 3)p^3$. The probability that yielded the greatest difference between the true generated number of triangles versus the expected number was 0.05, which had a difference of 690. There is a trend in the absolute value of the difference between the expectation and the generated number of triangles. The error between the expected number of triangles and the generated number increases with the edge inclusion probability. The rate of increase also increases with the probability. We have included a graph that tracks the increase in the error versus the edge inclusion probability. If we were to gather more data by continually incrementing the probability past 0.05 (to 0.06 and so on), we might find that the increase in the error is exponential with respect to the edge inclusion probability. However we can't make that assertion as we do not have enough data to claim that the increase is exponential. At a minimum, it is clear that the rate of increase in the error is also increasing.

Data (part 2)

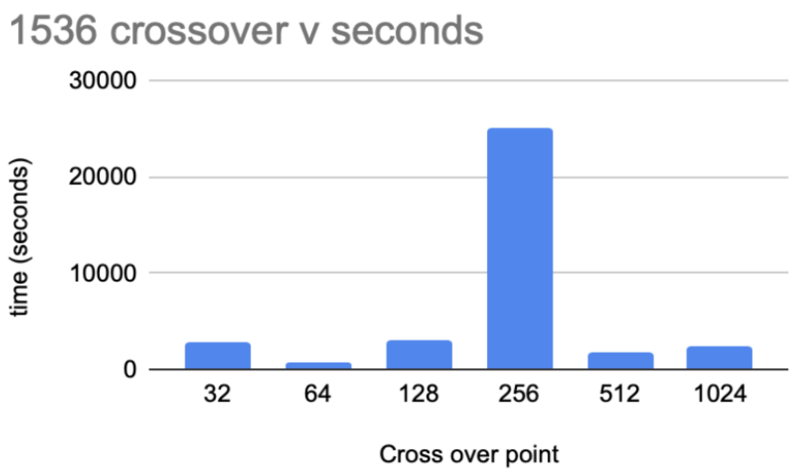
matrix size (d)	Crossover point	time (seconds)
512	32	16.561
512	64	16.047
512	128	17.673
512	256	19.290



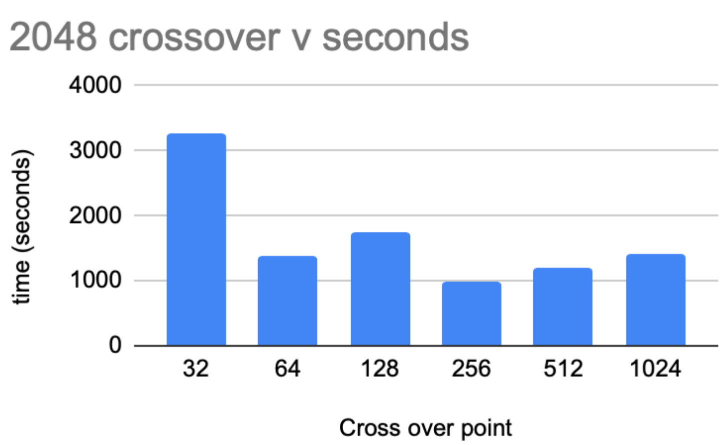
matrix size (d)	Crossover point	time (seconds)
1024	32	120.7579
1024	64	121.6589
1024	128	120.653
1024	256	127.139
1024	512	154.922



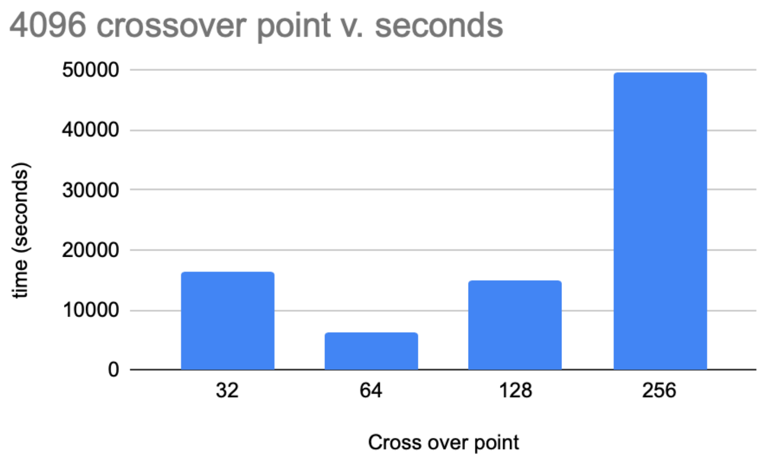
matrix size (d)	Crossover point	time (seconds)
1536	32	2857.050
1536	64	773.784
1536	128	3004.275
1536	256	25171.891
1536	512	1729.323
1536	1024	2522.717



matrix size (d)	Crossover point	time (seconds)
2048	32	3264.173
2048	64	1384.106
2048	128	1731.8754
2048	256	969.419
2048	512	1203.448
2048	1024	1416.738



matrix size (d)	Crossover point	time (seconds)
4096	32	16383.212
4096	64	6355.220
4096	128	14836.895
4096	256	49550.721



Data (part 3)

Graph Size (d)	Probability	Num Triangles	Expected Num Triangles	Abs value of difference
1024	0.01	180	178.433	1.567
1024	0.02	1410	1427.46	17.46
1024	0.03	4902	4817.69	84.31
1024	0.04	11627	11419.713	207.287
1024	0.05	22995	22304.128	690.872