

Pearson Cohen
CS-392 Assignment 4

Problem 1:

Matrix size	Block size	Time (milliseconds)
2048 x 2048	1	110.933
	2	53.876
	4	33.238
	8	27.778
	16	62.969
	32	63.321
	64	67.320
4096 x 4096	1	360.105
	2	197.913
	4	109.790
	8	96.412
	16	227.612
	32	232.413
	64	236.541
8192 x 8192	1	1532.934
	2	779.951
	4	461.726
	8	419.775
	16	937.380
	32	922.662
	64	927.863

A block width of 8 was consistently the fastest for this problem.

A cache hit occurs when the data being loaded for processing is already loaded into cache memory, thus making loading the data significantly faster. In this case, cache hits make the code faster because elements within the same “block” are near to each other in the 1D arrays storing the matrices. These nearby elements are cached, resulting in fewer cache misses and therefore faster execution. This is an example of spatial locality.

Problem 2:

Matrix size	Block size	Time (milliseconds)
2048 x 2048	1	172.037
	2	111.048
	4	60.145
	8	43.227
	16	38.092
	32	28.723
	64	30.496
4096 x 4096	1	759.457
	2	510.790
	4	231.688
	8	164.942
	16	127.961
	32	107.798
	64	108.901
8192 x 8192	1	3267.431
	2	2235.394
	4	977.301
	8	645.755
	16	576.963
	32	462.913
	64	468.761

A block width of 32 was consistently the fastest for this problem (by a small margin).

Problem 3: link to github repository:

<https://github.com/piercecohen1/CS392/commits/master>