Transp.c (Square Matrix)		
Matrix Width	Block Size	Time (Microseconds)
2048	8	38254
	16	35053
	32	32778
	64	40030
	128	47472
4096	8	194234
	16	155571
	32	132052
	64	176623
	128	216885
8192	8	772564
	16	706054
	32	593024
	64	622183
	128	1233968

Transp.c (Square Matrix)		
Matrix Width	Block Size	Time (Microseconds)
2048	8	41180
	16	38648
	32	31054
	64	39009
	128	53579
4096	8	208210
	16	194559
	32	149723
	64	160937
	128	223204
8192	8	741600
	16	662155
	32	583762
	64	699212
	128	1297576

https://github.com/Mmonaco13/CS392-HW4/commits/master

The code becomes faster because the blocks reduce the number of times the program has to go into the cache. If you do not use blocks, then the program has to go to cache far more often since every next value requires a large jump to the next value. Blocks helps minimize the number of large jumps that causes the program to have to go to the cache. The reason this issue is not as simple as going in the order of the array is because going in order for the input matrix is very out of order for the output matrix. Therefore, you want to find a region that is the most efficient.