UIN 1: 678523095				UIN 2: 656744567			
Name 1: Alex Tkaczyk				Name 2: Tameem Abdallah			

CS261 Machine Organization Homework 9 Template

Exchage Sort Algorithm				Quick Sort Algorithm			
Cache Size	Block Size	# Sets	Miss rate	Cache Size	Block Size	# Sets	Miss rate
4K	1	1024	97.8	4K	1	1024	15.5
4K	2	512	48.9	4K	2	512	7.8
4K	4	256	24.5	4K	4	256	3.9
4K	8	128	12.2	4K	8	128	2
8K	1	2048	91.5	8K	1	2048	11.9
8K	2	1024	45.8	8K	2	1024	6
8K	4	512	22.9	8K	4	512	3
8K	8	256	11.4	8K	8	256	1.5
16K	1	4096	66.4	16K	1	4096	7
16K	2	2048	33.2	16K	2	2048	3.5
16K	4	1024	16.6	16K	4	1024	1.7
16K	8	512	8.3	16K	8	512	0.9
32K	1	8192	6.6	32K	1	8192	3.3
32K	2	4096	3.3	32K	2	4096	1.7
32K	4	2048	1.6	32K	4	2048	0.8
32K	8	1024	0.8	32K	8	1024	0.4
64K	1	16384	0	64K	1	16384	2.7
64K	2	8192	0	64K	2	8192	1.4
64K	4	4096	0	64K	4	4096	0.7
64K	8	2048	0	64K	8	2048	0.3

Disscussion/ Comment / Conclusion	Disscussion/ Comment / Conclusion
Both cache size and block size have a inverse relationship with miss rate. When cache size increases, miss rate decreases. And when block size increases, miss rate decreases. When cache size was 64K, the block size did not matter as the miss rate became zero, meaning that eventually when the cache size crosses a certain boundary, the block size no longer matters.	Both cache size and block size have a inverse relationship with miss rate. When cache size increases, miss rate decreases. And when block size increases, miss rate decreases. Overall, Quick Sort had lower miss rates in comparison to exchange sort and the effects of cache size and block size changes are less significant (in comparison to XSort). In order for Quick Sort to reach a 0% miss rate, the cache size and block size would have to be increases significantly more in comparison to exchange sort.