CME213: Parallel Computing using MPI, OpenMP and CUDA
Homework 1: Radix Sort with OpenMP **Chi Zhang**
Stanford University SUID: 06116342
Spring 2018 Date: April 21, 2018

Problem 2: Radix Sort

	1	2	4	8	12	16	**24**	32	40	48
1	0.215	0.215	0.217	0.223	0.226	0.229	0.236	0.244	0.244	0.242
2	0.200	0.131	0.127	0.132	0.135	0.141	0.149	0.156	0.159	0.172
4	0.215	0.128	0.073	0.075	0.078	0.082	0.088	0.096	0.107	0.111
8	0.214	0.128	0.072	0.057	0.064	0.063	0.069	0.074	0.081	0.086
12	0.216	0.131	0.071	0.056	0.050	0.059	0.058	0.065	0.072	0.075
16	0.216	0.131	0.072	0.061	0.056	**0.048**	0.060	0.061	0.068	0.070
24	0.219	0.126	0.072	0.063	0.055	0.049	**0.048**	0.060	0.065	0.063
32	0.219	0.123	0.074	0.063	0.053	0.049	0.057	0.058	0.069	0.076
40	0.246	0.125	0.080	0.054	0.052	0.049	0.050	0.059	0.060	0.075
48	0.212	0.141	0.084	0.059	0.053	0.048	0.054	0.060	0.062	0.071

Table 1: Running times on `gpu-202-1` (row: thread number, column: block number)

- Compared to the serial version, the parallel radix sort runs over *four times* faster.

- For reasonable number of blocks, running time reduces around three times as the number of threads increase from 1 to 48. However, this is not the case when there are so few blocks. One extreme case is the first column, running time basically remains the same even though the number of threads varies, because a large number of threads are idle.

- With a fixed number of threads, increasing the number of blocks will typically improve runtime performance. However, if the number of block becomes unreasonably large, it will slow down computation. It is obvious there should be a balance between numbers of threads and blocks.

- As showed in Table 1, the best performance is achieved with 16 blocks/16 threads and 24 blocks/24 threads. Although numbers fluctuate in several runs, it seems these two combinations always yield the best performance. Given the fact that radix sort runs in parallel on 24 processors, the optimal option is 24 blocks/24 threads in theory, so that each core runs one thread computing one block.

- It can also be noticed that running a certain number of blocks with more number of threads does not help performance much. For instance, the running time using 16 blocks with 24/32/40/48 threads are almost identical, since only 16 threads are actually working in these cases.