Anderson Nguyen
EECS 120 HW1 Writeup

Starting Directory

```
Thu Apr 23 18:52:26 [3.29 3.36 3.50]  andersvn@hpc-login-1-3:~/EECS120-hw1
408 $ ls
driver.cc  LICENSE  Makefile  mergesort-omp.cc  sort.cc  sort.hh  sort.sh  timer.c  timer.h
```

After executing 'make mergesort-omp'

```
Thu Apr 23 18:55:34 [2.55 3.00 3.33]  andersvn@hpc-login-1-3:~/EECS120-hw1
411 $ make mergesort-omp
icpc  -openmp -O3 -g -o driver.o -c driver.cc
icpc  -openmp -O3 -g -o sort.o -c sort.cc
icpc  -openmp -O3 -g -o mergesort-omp.o -c mergesort-omp.cc
icpc -openmp -O3 -g -o mergesort-omp driver.o sort.o mergesort-omp.o

Thu Apr 23 18:55:42 [2.59 3.00 3.32]  andersvn@hpc-login-1-3:~/EECS120-hw1
412 $ ls
driver.cc  driver.o  LICENSE  Makefile  mergesort-omp  mergesort-omp.cc  mergesort-omp.o  sort.cc  sort.hh  sort.o  sort.sh  timer.c  timer.h
```

Sequential, N = 1
Quicksort Time = 1e-06 seconds
Mergesort Time = 0.109126 seconds

```
Fri Apr 24 01:52:21 [3.04 3.34 3.21]  andersvn@hpc-login-1-3:~/EECS120-hw1
467 $ ./mergesort-omp 1
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 1

Quicksort: 1e-06 seconds ==> 1 million keys per second
        (Array is sorted.)
My sort: 0.109126 seconds ==> 9.16372e-06 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 01:52:27 [3.28 3.38 3.22]  andersvn@hpc-login-1-3:~/EECS120-hw1
468 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492009 0.00000 mergesort- andersvn     qw    04/24/2020 01:51:57                                  8
```

Parallel, N = 1
Quicksort Time = 0 seconds
Mergesort Time = 0.061505 seconds

```
Fri Apr 24 02:04:22 [3.48 3.08 3.08]  andersvn@hpc-login-1-3:~/EECS120-hw1
486 $ ./mergesort-omp 1
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 1

Quicksort: 0 seconds ==> inf million keys per second
        (Array is sorted.)
My sort: 0.061505 seconds ==> 1.62588e-05 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 02:04:37 [3.36 3.07 3.08]  andersvn@hpc-login-1-3:~/EECS120-hw1
487 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492050 0.00000 mergesort- andersvn     qw    04/24/2020 02:04:22                                  8
```

Sequential, N = 1000
Quicksort Time = 0.000167 seconds
Mergesort Time = 0.076314 seconds

```
Fri Apr 24 02:05:10 [3.48 3.13 3.10]  andersvn@hpc-login-1-3:~/EECS120-hw1
489 $ ./mergesort-omp 1000
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 1000

Quicksort: 0.000167 seconds ==> 5.98802 million keys per second
        (Array is sorted.)
My sort: 0.076314 seconds ==> 0.0131038 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 02:05:24 [3.44 3.14 3.10]  andersvn@hpc-login-1-3:~/EECS120-hw1
490 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492050 0.00000 mergesort- andersvn     qw    04/24/2020 02:04:22                                 8
```

Parallel, N = 1000
Quicksort Time = 0.000147 seconds
Mergesort Time = 0.066347 seconds

```
Fri Apr 24 01:55:43 [2.97 3.20 3.18]  andersvn@hpc-login-1-3:~/EECS120-hw1
473 $ ./mergesort-omp 1000
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 1000

Quicksort: 0.000147 seconds ==> 6.80272 million keys per second
        (Array is sorted.)
My sort: 0.066347 seconds ==> 0.0150723 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 01:55:52 [2.97 3.19 3.18]  andersvn@hpc-login-1-3:~/EECS120-hw1
474 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492022 0.00000 mergesort- andersvn     qw    04/24/2020 01:55:43                                 8
```

Sequential, N = 10000000
Quicksort Time = 4.30543 seconds
Mergesort Time = 2.99191 seconds

```
Fri Apr 24 01:53:45 [3.26 3.34 3.22]  andersvn@hpc-login-1-3:~/EECS120-hw1
471 $ ./mergesort-omp 10000000
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 10000000

Quicksort: 4.30543 seconds ==> 2.32265 million keys per second
        (Array is sorted.)
My sort: 2.99191 seconds ==> 3.34235 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 01:54:17 [3.24 3.32 3.22]  andersvn@hpc-login-1-3:~/EECS120-hw1
471 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492012 0.00000 mergesort- andersvn     qw    04/24/2020 01:53:21                                 8
```

Parallel, N = 10000000
Quicksort Time = 2.79454 seconds
Mergesort Time = 2.83371 seconds

```
Fri Apr 24 02:01:55 [2.86 2.95 3.06]  andersvn@hpc-login-1-3:~/EECS120-hw1
483 $ ./mergesort-omp 10000000
Timer: gettimeofday
Timer resolution: ~ 1 us (?)

N == 10000000

Quicksort: 2.79454 seconds ==> 3.5784 million keys per second
        (Array is sorted.)
My sort: 2.83371 seconds ==> 3.52895 million keys per second
        (Array is sorted.)
        (Arrays are equal.)


Fri Apr 24 02:02:13 [2.92 2.96 3.06]  andersvn@hpc-login-1-3:~/EECS120-hw1
484 $ qstat -u andersvn
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2492044 0.00000 mergesort- andersvn     qw    04/24/2020 02:01:55                                    8
```

Therefore, **parallel merge is faster than sequential merge** through these examples. For my program, I recommend not going above 10 million values because it would take too long to sort. It makes sense that an array of size 10 million, quicksort and parallel merge sort have similar time complexities of O(nlogn). Quicksort is considered better because it is place sorting and does not need memory like mergesort. Merge sort would be better for something like a linked list or very large arrays. In my code, you can choose to either run it sequentially or parallel. I commented out the sequential function and it will parallel merge by default. I noticed that with small values of N, the difference in sorting time is not very large and sometimes could be sporadic because the value is so small and the hardware you use can also change the time.

Directory After Compiling

```
Fri Apr 24 02:08:42 [3.24 3.13 3.10]  andersvn@hpc-login-1-3:~/EECS120-hw1
493 $ ls
driver.cc  mergesort-omp          mergesort-omp.e2492022  mergesort-omp.o         mergesort-omp.o2492044  sort.hh  timer.h
driver.o   mergesort-omp.cc       mergesort-omp.e2492044  mergesort-omp.o2492009  mergesort-omp.o2492050  sort.o
LICENSE    mergesort-omp.e2492009 mergesort-omp.e2492050  mergesort-omp.o2492012  mergesort-omp.o2492052  sort.sh
Makefile   mergesort-omp.e2492012 mergesort-omp.e2492052  mergesort-omp.o2492022  sort.cc                 timer.c
```