Name : Sweta Rout
Unity ID : srout
Student ID : 200206715

Performance comparison: a table lists execution time, speedup (using mode 0 as the baseline) of each mode the excludes the data initialization and result verification.

I chose to implement merge sort.

Mode	Execution Time	Ratio to the base(mode 0)
0	16800041	1
1	7969001	0.47434414

What's the complexity of your algorithm?

Complexity of merge sort is **O(nlog n).** Parallel sorting using threads does not affect the time complexity. Because we perform same operations and share tasks to different threads.

Is the performance improvement as good as you expected? Why or why not?

As per the results mentioned above, running merge sort on a multicore processor is faster than running on a single core processor. This is because the data set is divided into different partitions and each part is given to a thread. After all partitions are sorted, they are merged. So working in parallel can save time.

Comparing your performance with your board mate, is your algorithm faster than theirs or not? Why or why not?

We have only 2 people in our group.

Name	Type of Sorting	Execution Time	Ratio to the base(mode 0)
Sweta Rout(Myself)	Merge Sort	7969001	0.47434414
Cody Nesbitt	Bucket Sort	32258621	1.8

Performance is based on algorithm and my algorithm has snippets that can be run independently with threads and my code is more optimized. His code may be having other overhead tasks that can't be done independently.

We are only 2 people, so I discussed with other students and came to know that bitonic sort has performance ratio of 1.8

The list of students sharing the same board with you :

- Sweta Rout (srout@ncsu.edu)
- Cody Nesbitt (cznesbit@ncsu.edu)