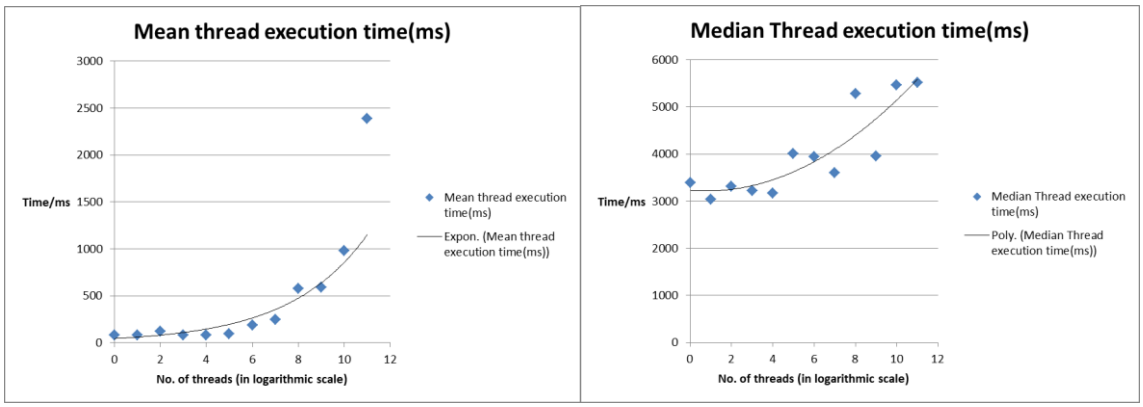
Woong Wen Tat (1002323)

The MeanThread.java runs by creating threads and waiting for each to end and prints the mean in each thread.

The MedianThread.java runs mergesort on each thread and then merges the results log2(number of threads) time using mergesort merge algorithm.

Discussion:



Mean thread execution time seems to be exponentially increasing with the number of threads, this results make sense as we have to print the mean execution time for each thread before combining all the threads into one. Hence, as the number of threads increases , we have to loop through more threads and the print overhead increases. The trendline is exponential because the x-axis is in a logarithmic scale.

Median thread execution time is not consistent, but the overall trend is exponentially increasing. Merging of the threads is O(number of threads) and merging 2 threads is O(2*thread length) so it should be O(thread length) per thread. The total time for merging should be constant O(number of thread*thread length) but as number of threads increases we need to do merging for a log2(number of threads) time, which means that the median execution time will increase linearly with number of thread.