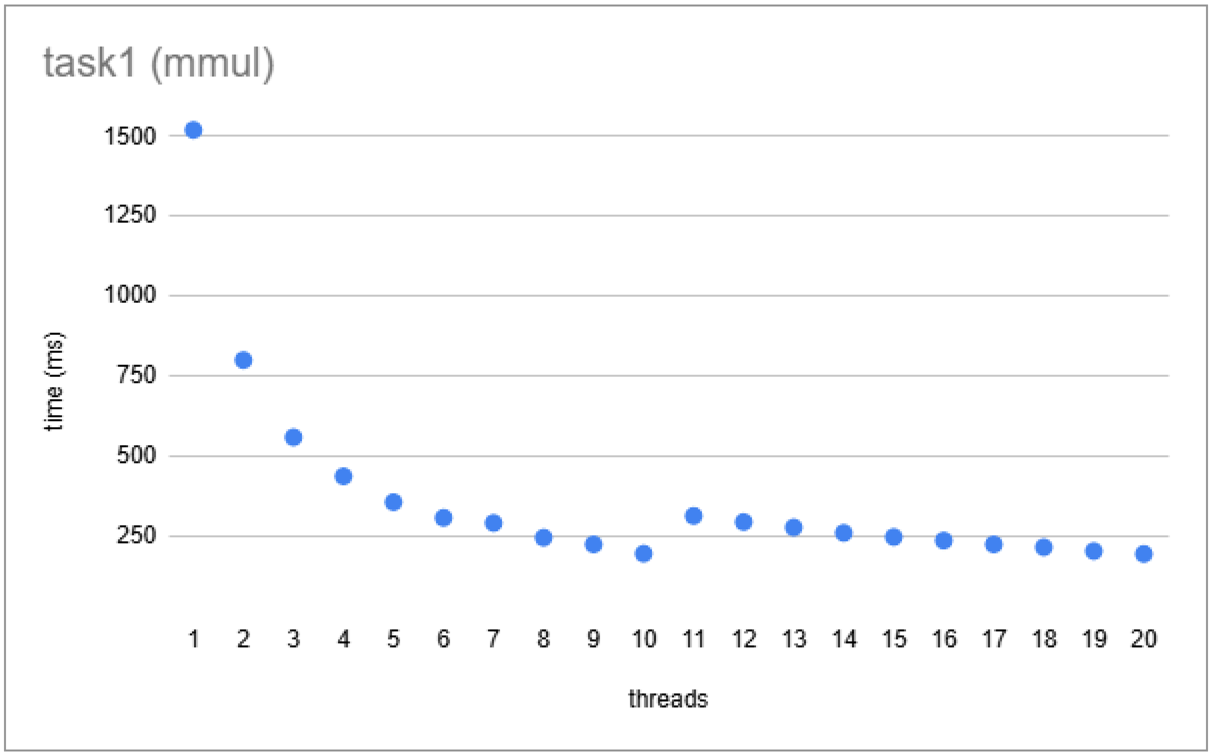
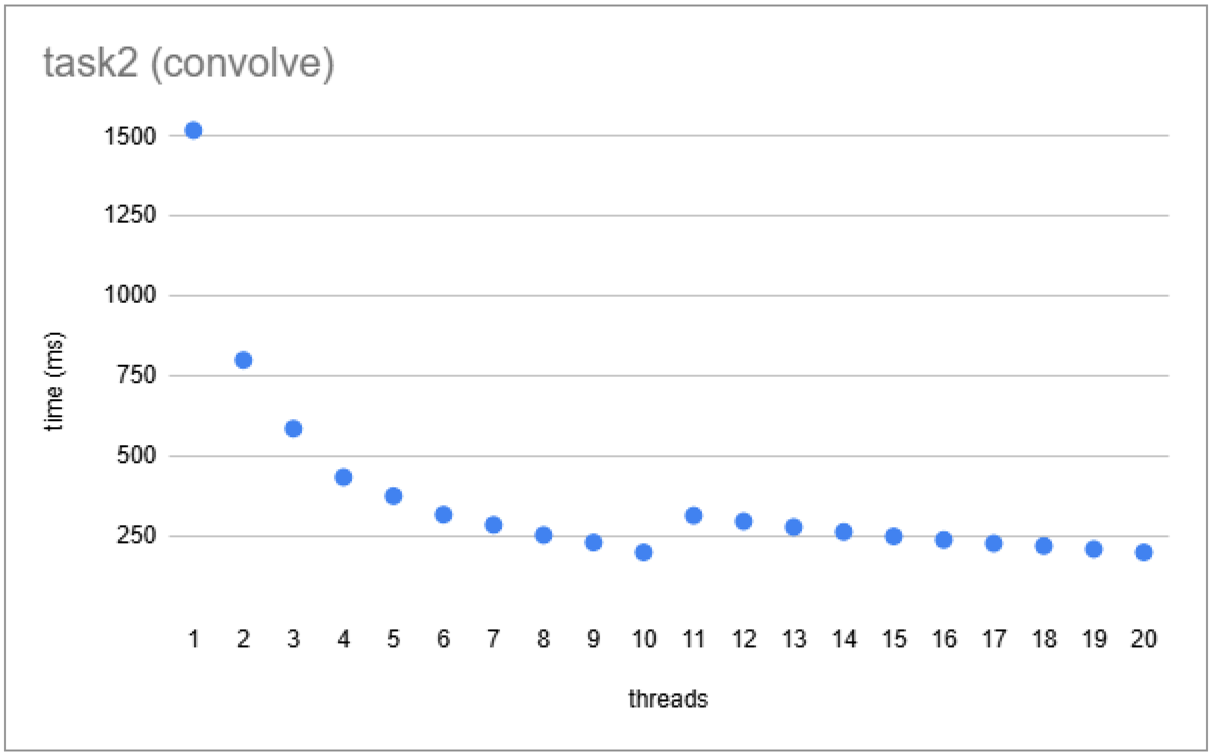
Leah Blasczyk
HW02
GitHub Username: **Lebrra**
(Assignment 3 Path: https://github.com/Lebrra/repo759/tree/main/HW03)

task1 plot:

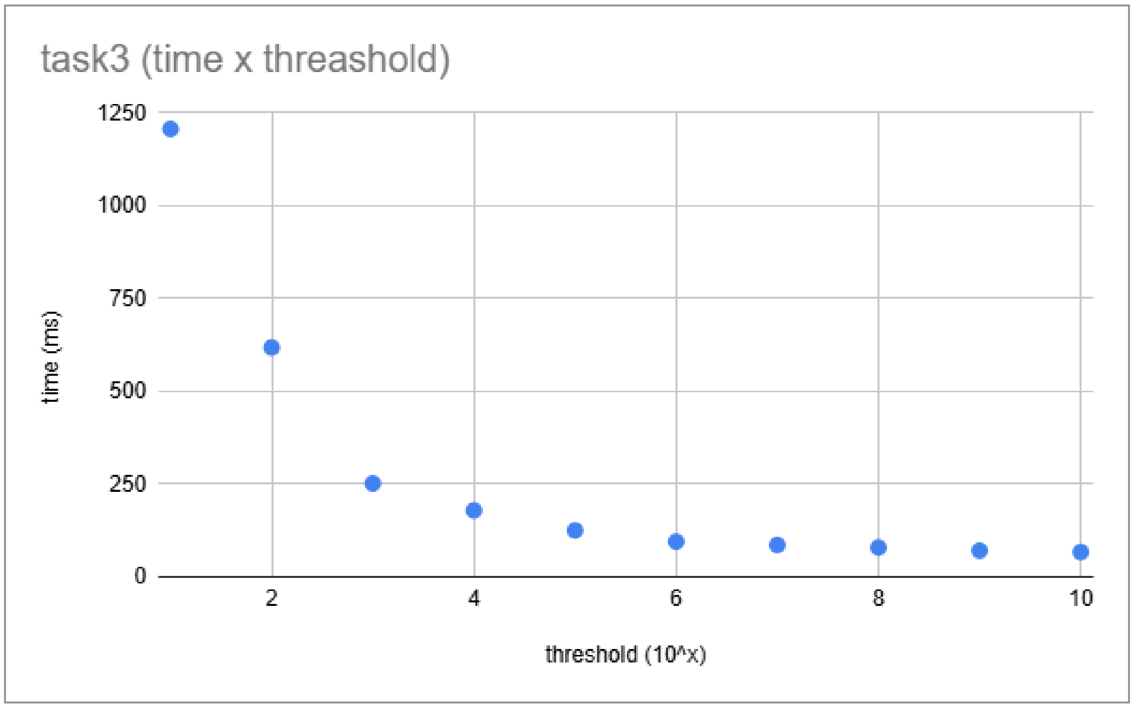


task2 plot:

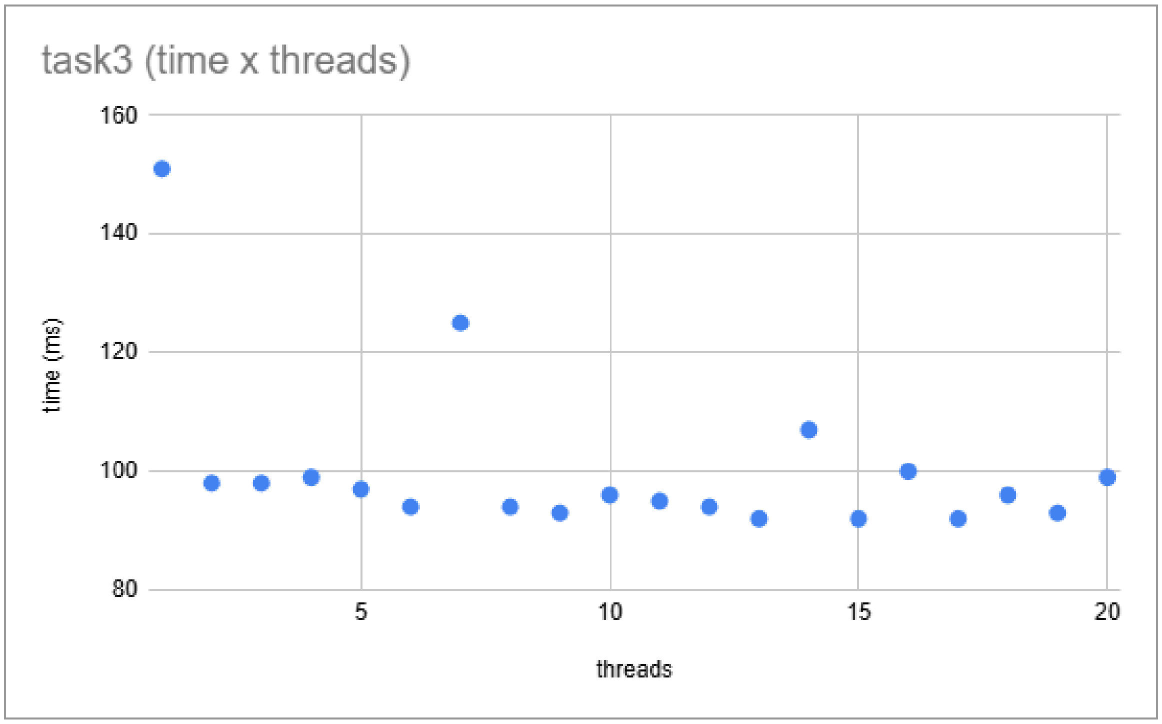


It is true that utilizing more threads gives faster results, but the benefits start to even out around 10 threads. As you can see, the algorithm actually gets a bit slower between 10 and 11 threads. This is likely the point where the effort of managing all the threads is becoming more taxing than the algorithm that is being computed.

task3 plot (threshold)



task3 plot (threads)



Notice that there isn't much gain in time using multiple threads in mergesort. This makes sense to me because merge sort is dependent on its past recursion steps (at least how I implemented it) and therefore requires a lot of gaiting in implementation. This is also why a higher threshold value => less recursion causes a quicker run time.

****Also note that the threads plot for task3 can be run with task3_2.sh**