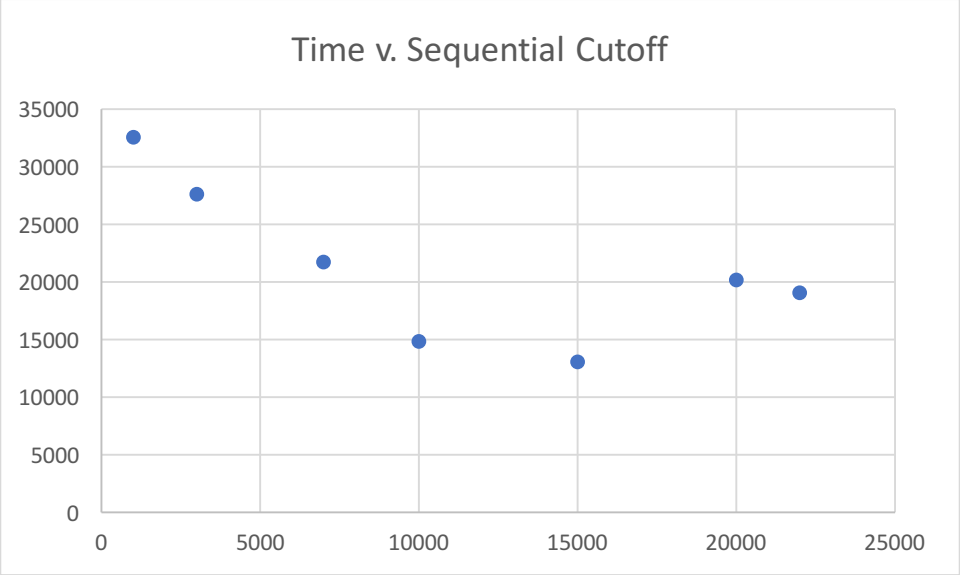
Elvis Kahoro (ekahoro) and Ethan Hardacre (ehardacre)

How we split up the work: We worked together the entire time on one computer. We switched off periodically.

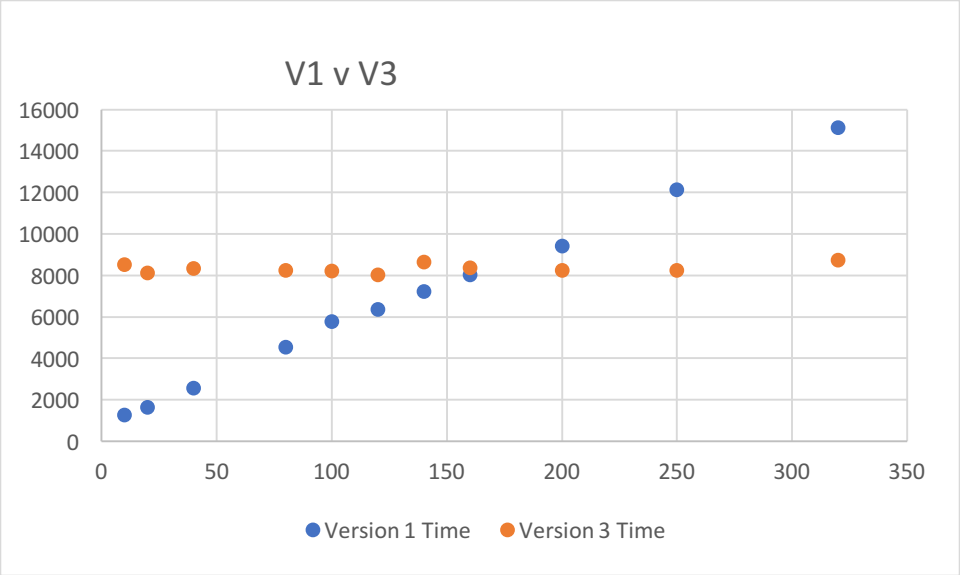
We tested the program as we built it by using the debugger, system out statements and the occasional Junit when we knew we were over our heads with testing a particular method. We considered all of the boundary cases. Smaller inputs, we used 10 by 10 grids so that we could check our population consistency between each version.

We used the mac desktops in the computer lab. The computers have 4 physical cores.

Sequential cut-off vs time. The time for running the program, decreased until 15000. The sequential cut off at values very close to the size of the lists led to an increase in timing, we think this is because the two threads are created to process about 21000 pieces of information, but one thread works on 20000 and the second thread runs a couple of hundred which is an uneven distribution of tasks. We want the threads to be manipulating about the same amount of information



Parallelism did not help because we do not have enough cores to make it efficient and we also aren't dealing with that much data. The graph below shows our queries and how they changed. Paralellism was only helpful once we got over 150 queries.



We did not have any extra projects.