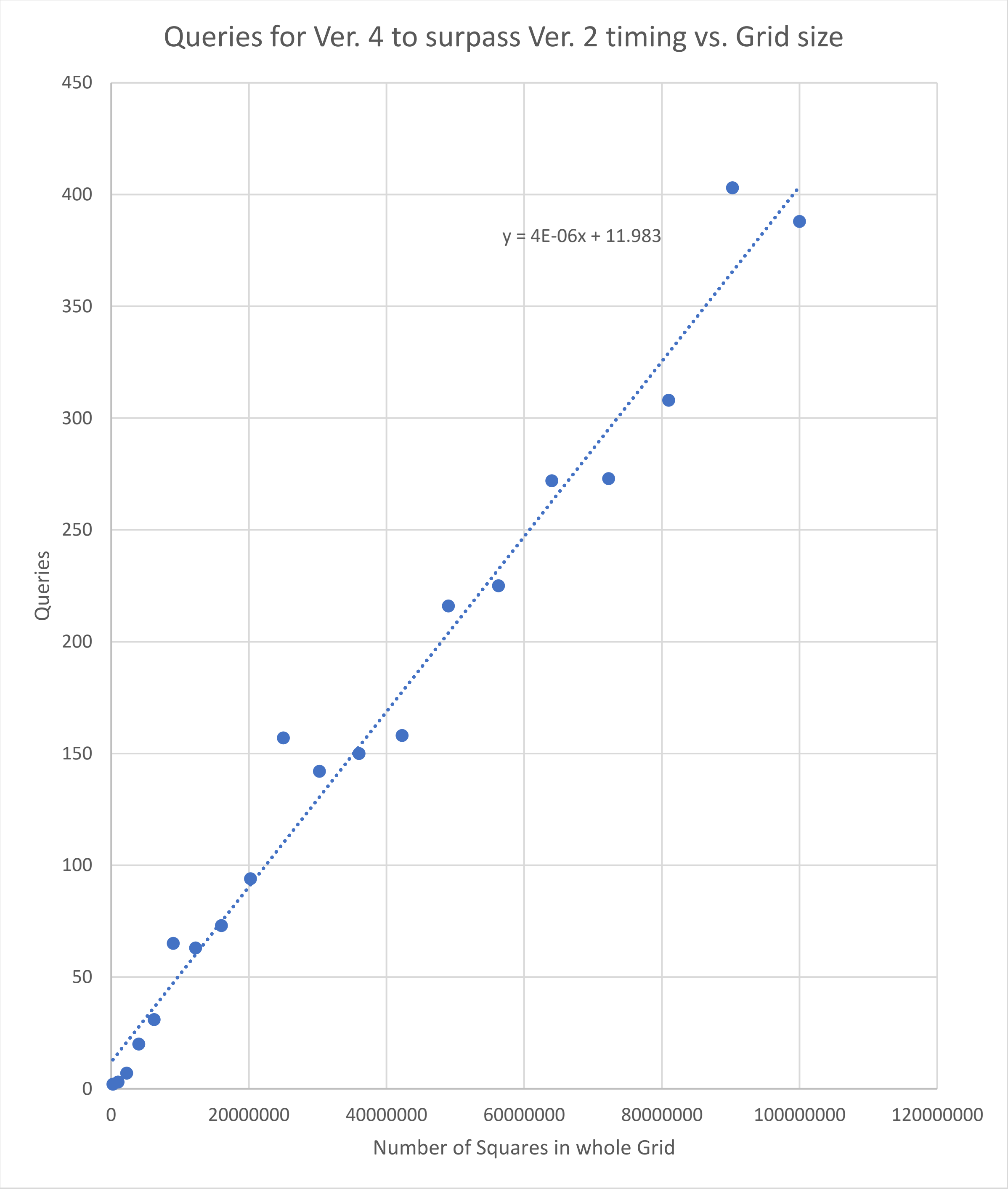
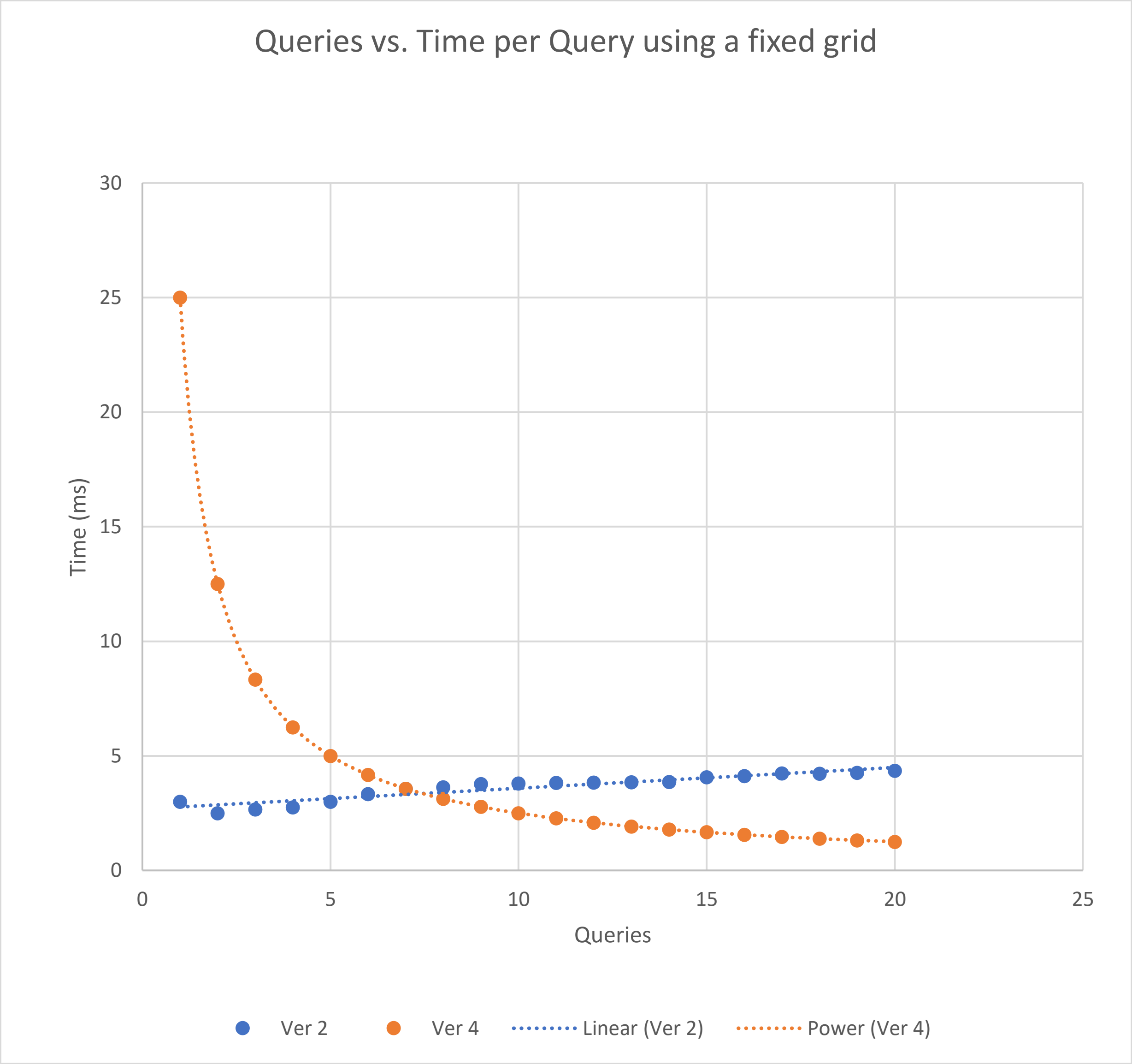


Number of Queries Ver. 4 must at least do to make the preprocessing worth it over Ver. 2. Compared to the comparison of Ver. 3 and Ver. 1, it seems like as size of the grid increases, the rate of which the queries it takes for the preprocessing to be "worth it" is greater for Ver. 4 than it is for Ver 3. This probably is because the Fork Join framework takes a lot of processing to do in managing and creating threads, but for much much larger grids it would probably be the opposite. All queries were generated randomly and they were based upon the size of the grid. Before gathering data, there was a warmup in which both versions were run 100 times.



The average amount of time each query as amount of queries go up. Ver. 4 gets a similar average time per query around the 7th query, which is around the time in which Ver. 4's preprocessing time becomes "worth it" over Ver. 2. This was done with a fixed array of 1000000 squares (1000X1000). There seems to be a linear relationship between Ver. 2's queries, and there seems to be an inverse relationship between Ver. 4's queries and the average time of the queries in that as the queries increase, the average time it takes one query approaches 0. Comparing this to the comparison of Ver. 3 and Ver. 1, we can see that it takes more queries to make Ver. 4 worth it compared to Ver 2 that it does Ver. 3 and Ver. 1, and that the line of Ver. 4 does not approach 0 as fast as it did in Ver. 3. This is probably because there is a significantly larger preprocessing time for Ver. 4 for this size grid, given that the Fork Join Framework has to create and manage threads, than there was in Ver. 3. A much larger grid may show that Ver. 4 becomes "worth it" in less queries than Ver. 3.