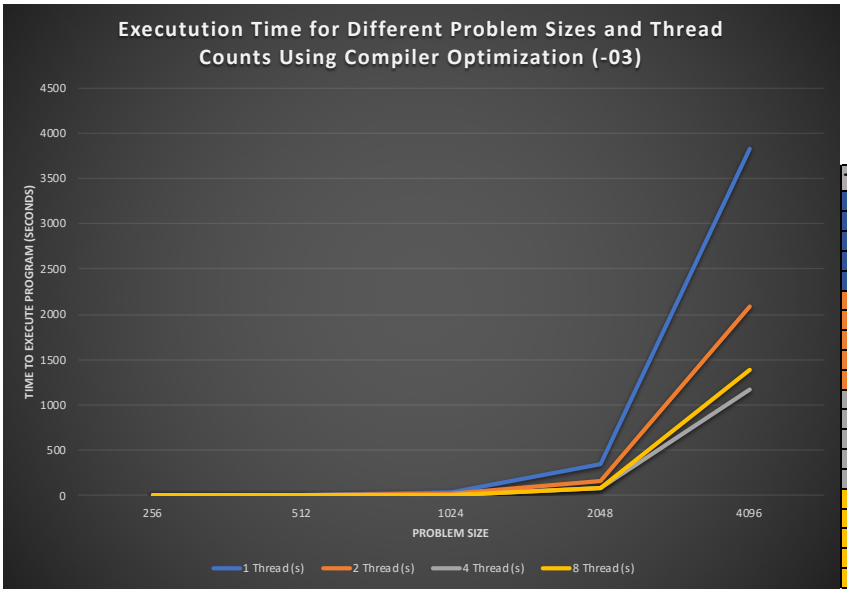
Jacob Lipton
CPSC 3300
October 2, 2019

Task B

1. 

Threads	Problem Size	Time
1	256	0.111465
1	512	0.894959
1	1024	30.847561
1	2048	343.96974
1	4096	3826.88244
2	256	0.05545
2	512	0.44889
2	1024	15.28629
2	2048	166.322503
2	4096	2091.9701
4	256	0.02933
4	512	0.238031
4	1024	8.015914
4	2048	85.605506
4	4096	1171.92778
8	256	0.033235
8	512	0.229698
8	1024	8.217266
8	2048	81.616454
8	4096	1383.17592

2. For one thread, the theoretical execution time is 32.49112109 seconds and the actual time elapsed was 32.5108502, so these values match when you take into account the overhead that *perf* added.
3. The number of instructions is only different by about 3.5 billion, which is not that significant. This makes sense though, because even with more threads the same computations would have to be made, even if parallel.
4. The theoretical maximum speedup would be 4 times as fast as the 1 threaded program. So in this case, the theoretical maximum speedup time for the 4 threaded program would be an execution time of 8.127712551. Somehow, the 4 threaded example was faster than that, with an execution time of 8.019936906 seconds.
5. Technically we got 101.34% of peak performance.
6.

Compiler Flag	Cycles	Instructions	Clock Rate (GHz)	Time Elapsed (s)
-g	165,632,429,042	182,926,461,058	3.792 GHz	11.11170572
-0	157,357,273,416	182,928,393,343	3.792 GHz	10.55133718
-1	147,926,499,889	182,920,073,832	3.791 GHz	9.758074861
-2	155,986,967,578	182,926,673,553	3.791 GHz	10.29656489
-3	147,284,329,542	182,913,278,630	3.791 GHz	9.715429997

7. Optimizations are not turned on by default because it keeps each statement independent of each other to make it easier for debugging purposes. They are not turned on by default because typically the increased performance is at the expense of compilation time and the ability to debug. At -01 ***-fauto-inc-dec*** is turned on, which combines memory addresses that have similar increments or decrements to the same memory accesses. Also ***-fguess-branch-probability*** is turned on, which tells the compiler to not use heuristics to guess branches in the program. At -02, ***-falign-loops*** is turned on, which will make loops loop

to a power of 2, optimizing them to caches, as well as ***-ffinite-loops*** being turned on, which makes the compiler assume that all loops will eventually exit, and not loop infinitely. At -03 ***-ftree-loop-vectorize*** and ***-ftree-vectorize*** are turned on, which turn all trees and loops into vectors for faster processing.