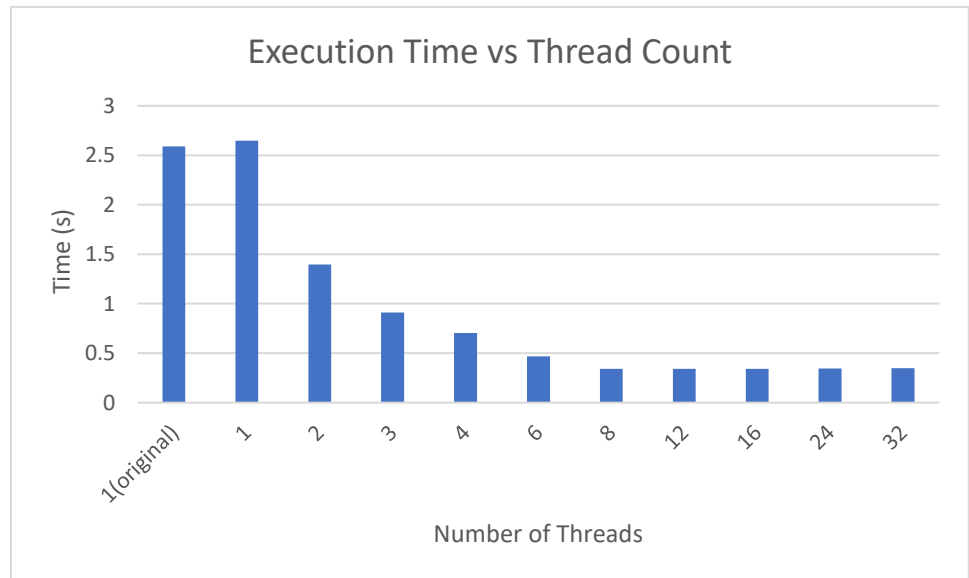


Question 2:

A.

Threads	Timing(s)
1(original)	2.591
1	2.648
2	1.397
3	0.911
4	0.705
6	0.469
8	0.343
12	0.343
16	0.342
24	0.344
32	0.347



B. No, I do not observe N times speed up when compared to the original single threaded program.

C. The reason we stop seeing a speed up is because there is some other inherent bottleneck occurring stopping us from reaching faster speed. Where something is slowing down the rest of the threads from being run in parallel. For example, in this case it might be that since there is a limit to the number of cores on the CPU it can't context switch fast enough between the threads to run them in parallel. We're aware that the linux machines have 8 cores or at least the 9700 cpu that this was tested on does. And that goes hand in hand with when the speed stopped becoming faster.

Question 4:

Medium.txt

#Threads	Timing(s)	Observed Speedup compared to original	Expected Speedup
Original	17.4604	1	1
1	17.1831	1.016137949	1
2	9.0329	1.932978335	2
3	6.0265	2.897270389	3
4	4.6029	3.793347672	4
8	2.2648	7.70946662	8
16	2.3483	7.435336201	16

Hard.txt

#Threads	Timing(s)	Observed Speedup compared to original	Expected Speedup
Original	5.883	1	1
1	5.8719	1.001890359	1
2	3.0931	1.901975365	2
3	2.0535	2.864864865	3
4	1.7108	3.438742109	4
8	0.768	7.66015625	8
16	0.7717	7.623428794	16

Hard2.txt

#Threads	Timing(s)	Observed Speedup compared to original	Expected Speedup
Original	6.0078	1	1
1	5.9534	1.009137636	1
2	3.0696	1.95719312	2
3	2.0917	2.872209208	3
4	1.5653	3.8381141	4
8	0.7664	7.838987474	8
16	0.7566	7.940523394	16

The timings all match up with the expected speed up until the 16th thread run. This is however still expected behaviour as mentioned previously the CPU this was tested on (i7-9700) has 8 cores so it can concurrently run 8 threads without having to context switch between this program. So we can expect each thread to be run on a single core and since we're past that limit after 8 we observe diminishing returns if any at all.