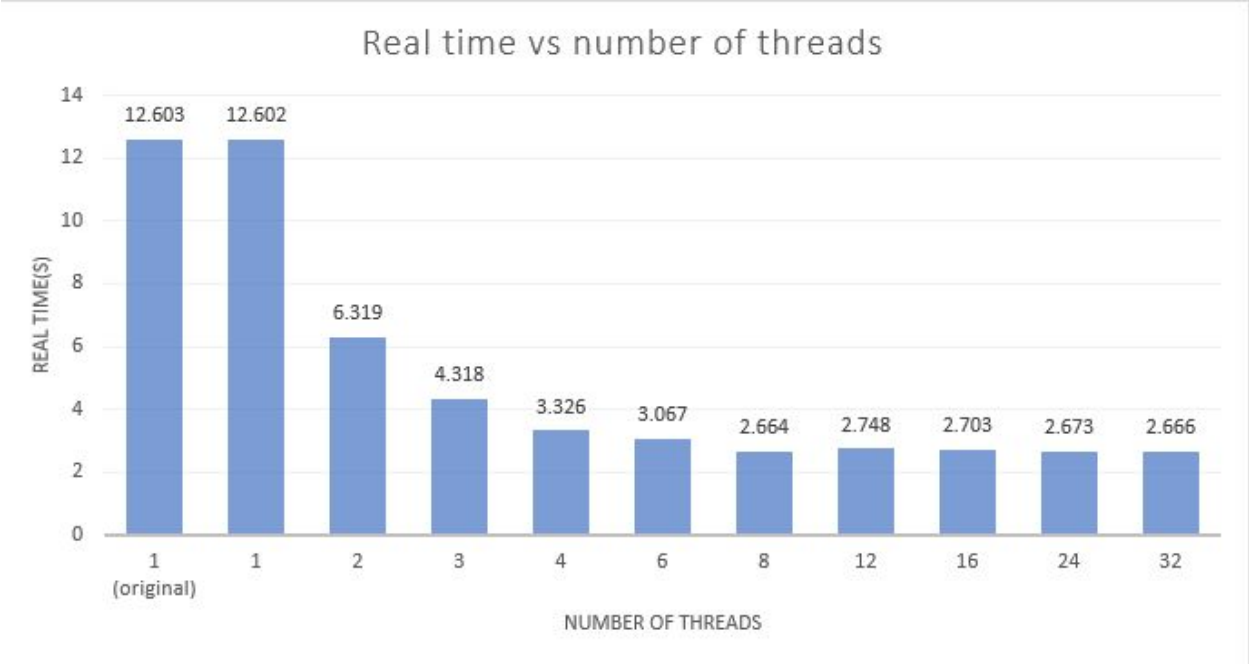
2.

Threads	Timing(s)
1 (original)	12.603
1	12.602
2	6.319
3	4.318
4	3.326
6	3.067
8	2.664
12	2.748
16	2.703
24	2.673
32	2.666



a.

With N threads you should see N-times speed up compared to the original single threaded program. Do you observe this in your timings for all N?

No, in the beginning in thread 1,2,3,4 we observe the expected n time speed up compared to original but starting from thread 6, the timing does not have a N-times speed up compared to the original single threaded program.

b.

Why do you stop seeing the speed up after some value of N?

This is because we can only have c number of improvements, where c is the number of cores. Thus, once all of them are occupied, increasing the number of threads will not make the operation faster as they will have to wait for the core to complete their existing tasks before being processed in parallel.

Q4.

	Test file: medium.txt		
#threads	Observed timing	Observed timing comapred to original	Exepted speedup
original program	18.8	1	1
1	19.4	0.968	1
2	9.95	1.89	2
3	6.65	2.83	3
4	5.1	3.69	4
8	3.74	5.03	8
16	3.88	4.86	16
	Test file:hard.txt		
#threads	Observed timing	Observed timing comapred to original	Exepted speedup
original program	6.43	1	1
1	6.61	0.971	1
2	3.39	1.89	2
3	2.26	2.84	3
4	1.74	3.68	4
8	1.27	5.07	8
16	1.27	5.05	16
	Test file: hard2.txt		
#threads	Observed timing	Observed timing comapred to original	Exepted speedup
original program	6.41	1	1
1	6.61	0.971	1
2	3.39	1.89	2
3	2.29	2.8	3
4	1.75	3.67	4
8	1.26	5.07	8
16	1.28	5	16

The observed timing compared to original is less than the expected speedup because we only have a limited amount of cores to process our program. So any more thread than the number of cores, we cannot process them as these threads at the same time as the hardware only had 4 threads. So, for more threads than cores. will have to wait for the cores to finish the existing threads before they can process these threads in parallel which causes the time to not increase as expected. Also, because thread, barrier, mutex initialization, which did not occur in the original, takes some time, the code will run a little slower, so we do not get the increase in performance that was expected.