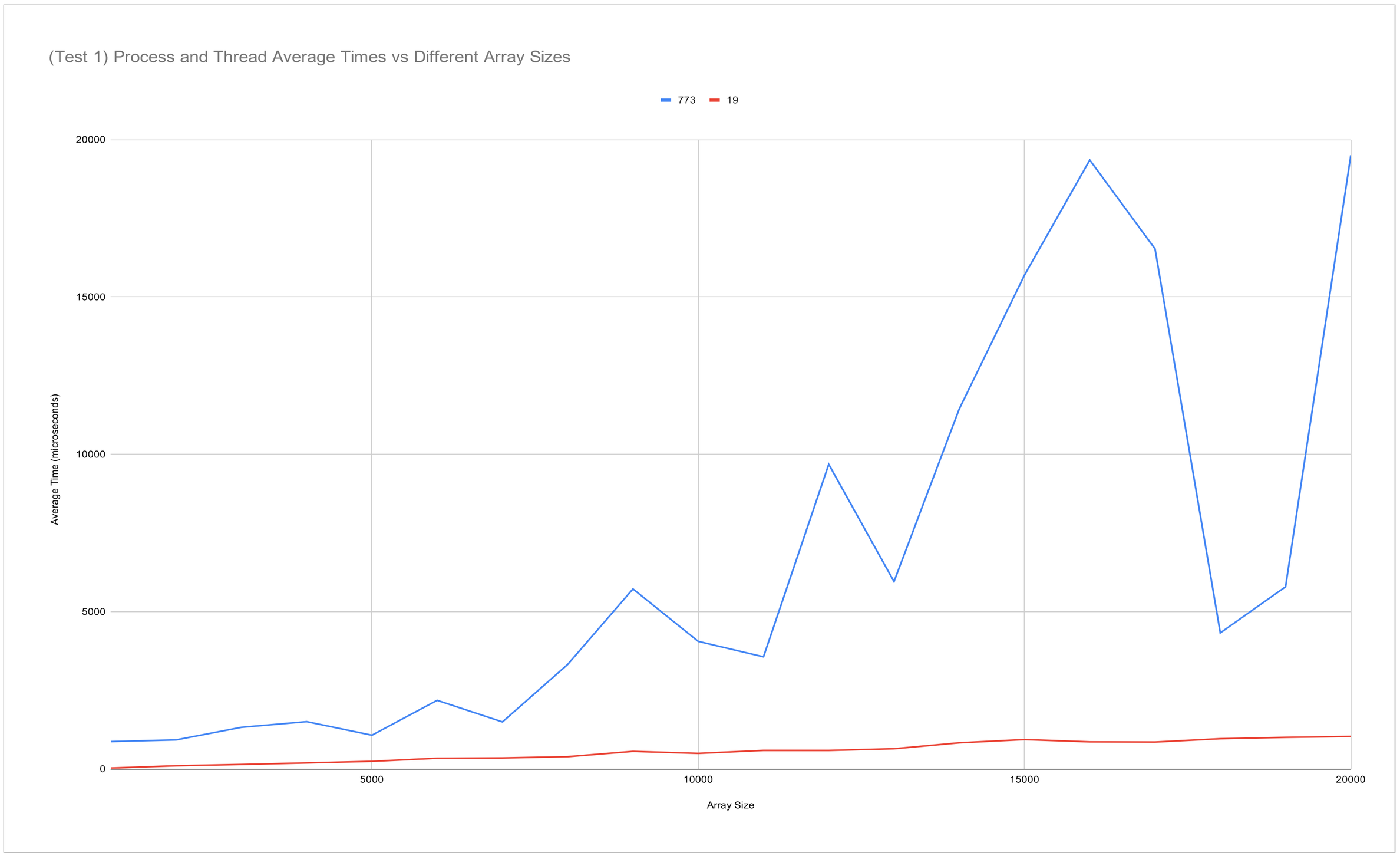
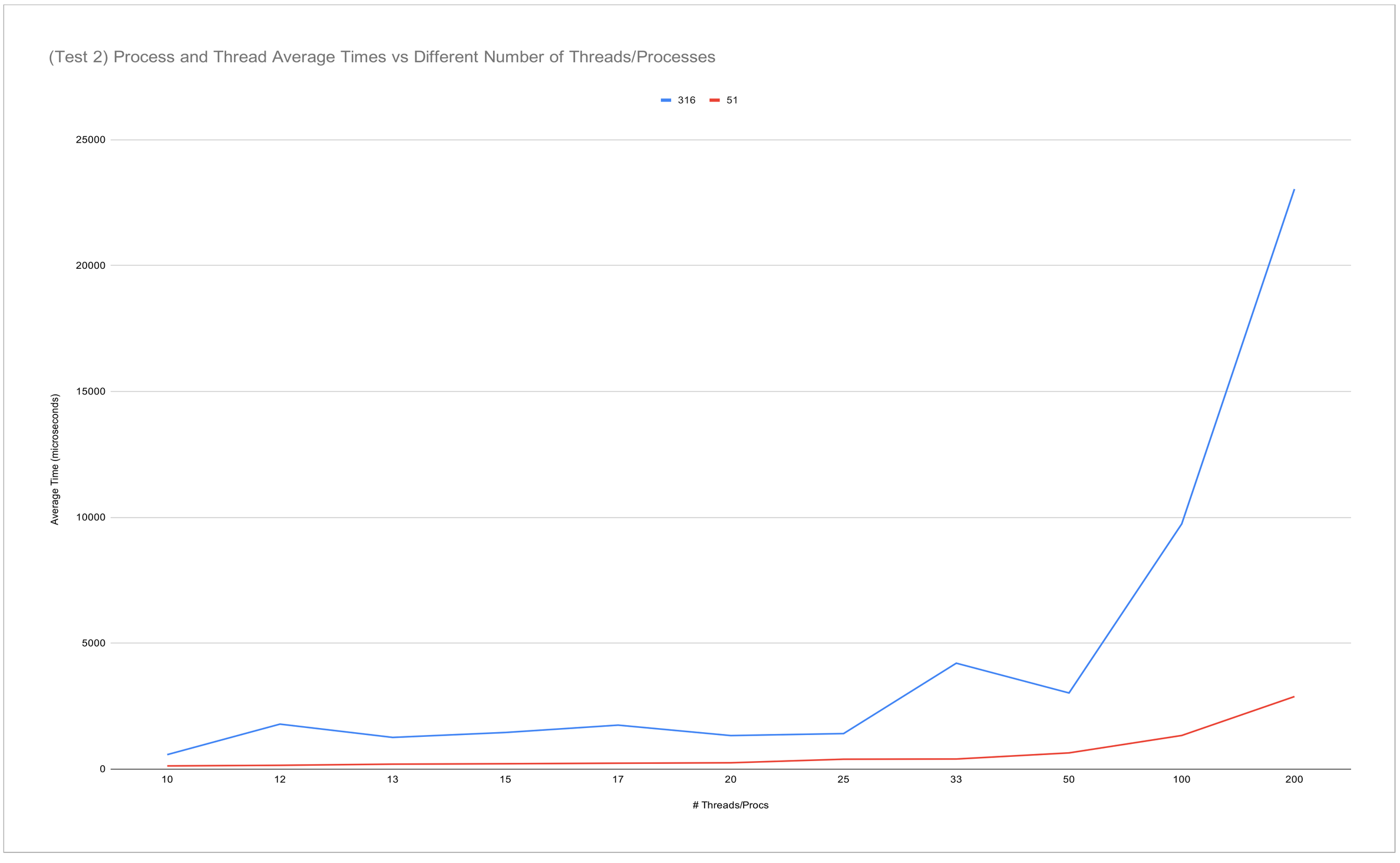
Test 1											
Partition Size	Array Size	# Threads/Procs	Proc Avg. Time	Min	Max	Std. Dev.		Thread Avg. Time	Min	Max	Std. Dev.
100	**100**	1	**256**	85	620	208		**17**	11	312	32
250	**500**	2	**773**	457	1307	258		**19**	14	241	22
250	**1000**	4	**872**	120	2488	602		**31**	23	397	37
250	**2000**	8	**925**	91	4797	878		**103**	77	757	67
250	**3000**	12	**1325**	84	6870	1501		**145**	122	749	61
250	**4000**	16	**1505**	81	8416	1822		**194**	164	993	82
250	**5000**	20	**1074**	87	3183	617		**244**	216	856	64
250	**6000**	24	**2183**	89	10698	2129		**342**	255	3044	314
250	**7000**	28	**1496**	188	3204	812		**351**	304	1352	108
250	**8000**	32	**3324**	90	19355	3654		**392**	345	1427	108
250	**9000**	36	**5723**	232	12413	3319		**561**	418	1909	197
250	**10000**	40	**4056**	90	27335	4660		**497**	446	1283	85
250	**11000**	44	**3565**	87	24763	4377		**591**	487	3988	401
250	**12000**	48	**9683**	203	31701	9019		**590**	547	1198	66
250	**13000**	52	**5957**	88	32717	7114		**644**	594	1182	69
250	**14000**	56	**11443**	218	37969	11079		**833**	638	4866	459
250	**15000**	60	**15698**	220	41576	12850		**936**	588	1988	260
250	**16000**	64	**19353**	425	43472	13144		**863**	741	5859	513
250	**17000**	68	**16530**	554	47445	14376		**857**	784	1392	73
250	**18000**	72	**4326**	80	9146	2635		**963**	832	6192	544
250	**19000**	76	**5791**	88	47884	5325		**1006**	892	2744	211
250	**20000**	80	**19505**	576	57663	16272		**1035**	933	2023	113
Test 2											
Array Size	# Threads/Procs	Partition Size	Proc Avg. Time	Min	Max	Std. Dev.		Thread Avg. Time	Min	Max	Std. Dev.
1000	**4**	250	**277**	94	527	120		**35**	24	588	55
1000	**5**	200	**316**	95	804	161		**51**	37	533	49
1000	**10**	100	**569**	95	1056	303		**121**	95	668	55
1000	**12**	83	**1781**	487	5988	1407		**145**	118	842	71
1000	**13**	76	**1253**	96	1850	350		**191**	140	852	90
1000	**15**	66	**1449**	97	2374	435		**207**	179	803	61
1000	**17**	58	**1741**	91	2887	426		**229**	174	1337	127
1000	**20**	50	**1326**	91	2914	743		**245**	210	724	54
1000	**25**	40	**1405**	92	3124	833		**387**	262	2298	212
1000	**33**	30	**4202**	1728	7007	1194		**395**	357	1147	79
1000	**50**	20	**3019**	95	6371	1636		**638**	575	1253	72
1000	**100**	10	**9736**	99	26416	6407		**1330**	1196	3590	270
1000	**200**	5	**23038**	354	42918	12217		**2876**	2601	8698	626


(Test 1) Process and Thread Average Times vs Different Array Sizes
773
19
20000
15000
10000
5000
0
Average Time (microseconds)
5000
10000
15000
20000
Array Size


(Test 2) Process and Thread Average Times vs Different Number of Threads/Processes
316
51
25000
20000
15000
10000
5000
0
Average Time (microseconds)
10
12
13
15
17
20
25
33
50
100
200
Threads/Procs

Test 3	Array Size	# of Thread/P	artition Size	verage Time
Thread	250	8	31	120
Process	250	1	250	119
Thread	1000	80	12	1455
Process	1000	10	100	1575
Thread	10000	1600	6	28812
Process	10000	200	50	26321
Thread	20000	4000	5	74462
Process	20000	500	40	105461
Thread	40000	4000	10	75171
Process	40000	500	80	109759

Test 1 runs 22 times with increasing array size from 100 to 20000 with the same partition size of 250, exclude the array of size 100. The graph to the right looks at the average runtime for each test, with the blue line representing processes and red representing threads. You can clearly see that processes tend to have a higher average runtime than threads and as the array size gets bigger and bigger the difference between average times becomes greater.

Test 2 consists of running the search with an array size of 1000 but the number of threads or processes increases. Once again the data shows processes take more time searching an array than threads and the time difference increases as the number of threads or processes used increases.

The purpose of **Test 3** is to estimate how many threads it would take to equal the time it takes for processes to find a target within an array. After running through multiple tests we found that it takes 8 - 10 threads to equal search time of one process