Assignment 2: Spooky Searching

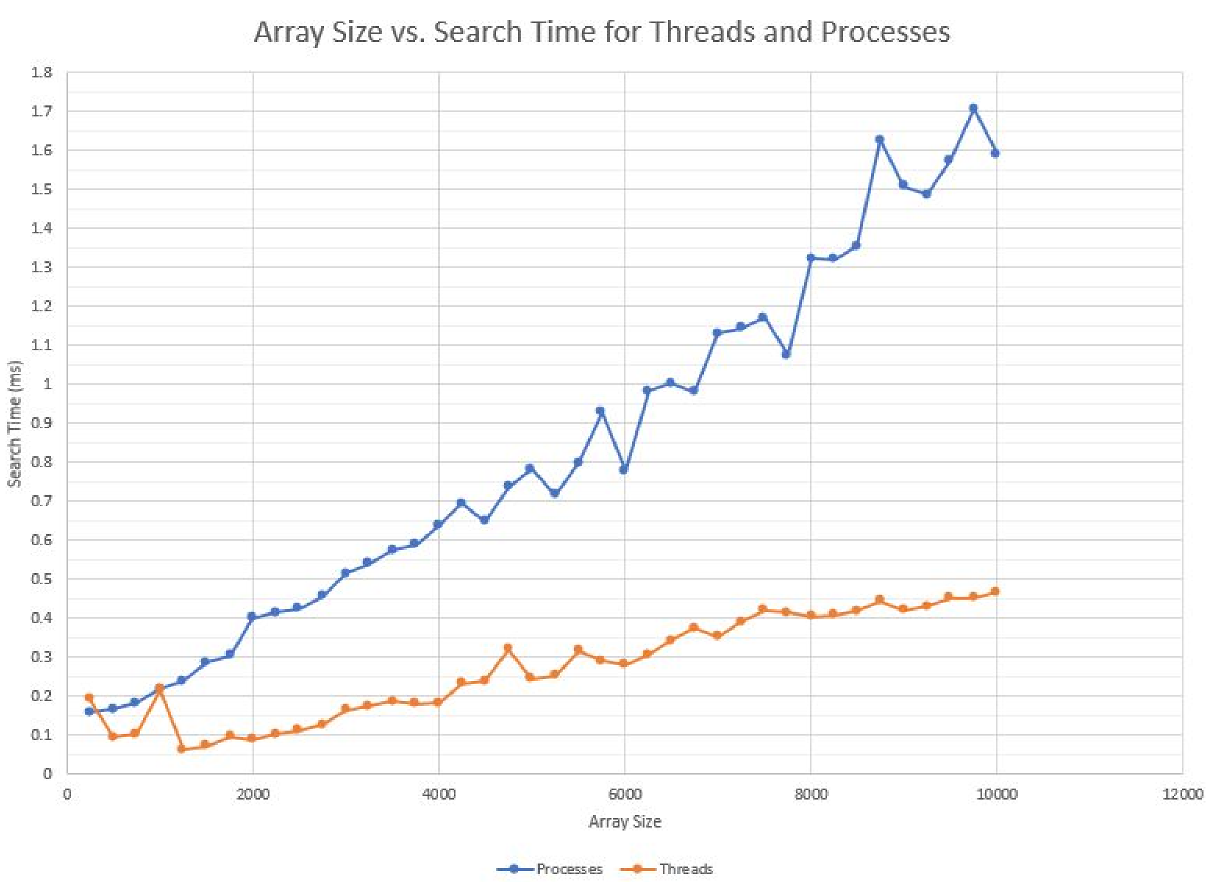
Daniel Bezden and Param Thakker

Test A

We tested the time it took for two different algorithms to search for a value in an unsorted array, one using multiple processes and the other using multiple threads. We found that the search time generally increased linearly, indicating a O(n) time complexity. Threads typically performed the search in about one third of the time of processes.

Best fit equation for processes: t = (160.41 ns)n
Best fit equation for threads: t = (50.2121 ns)n

Figure 1: Results from Test A



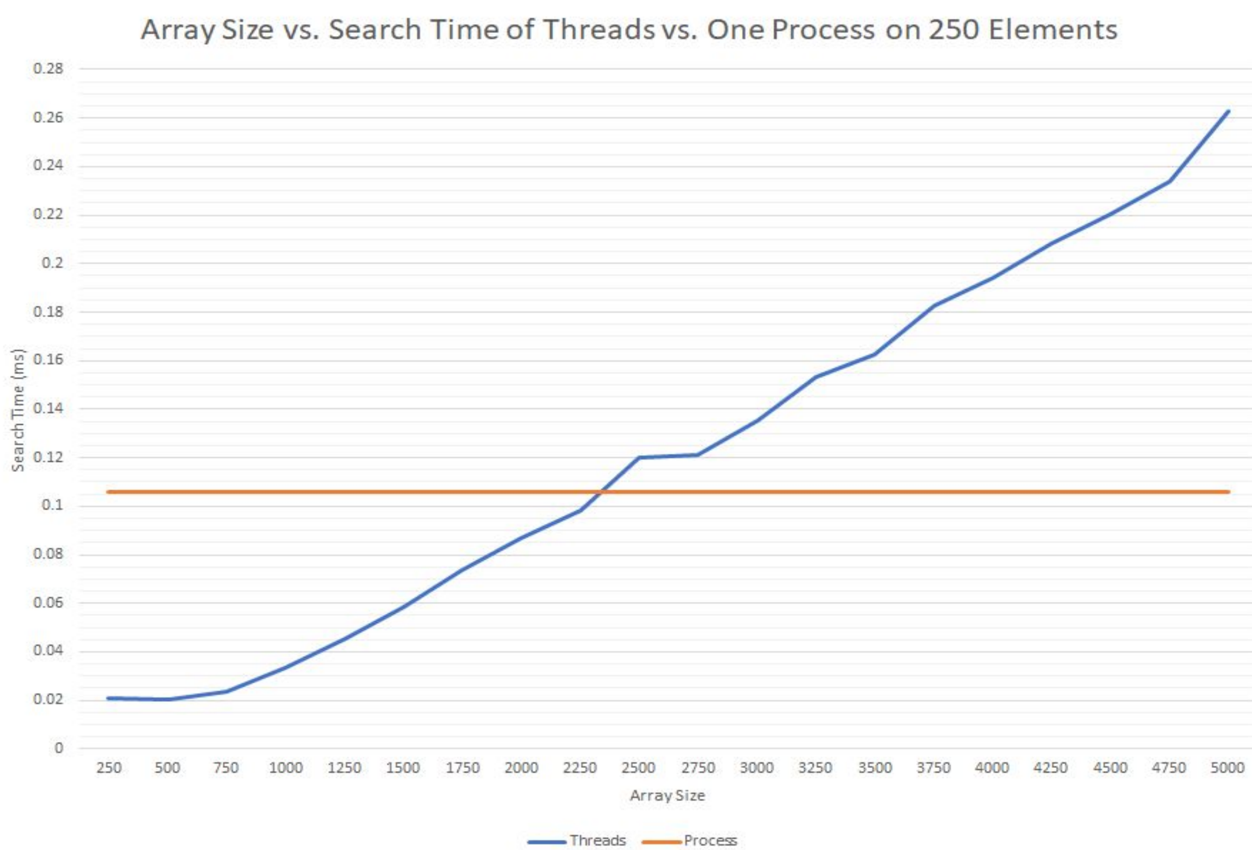
Data:

Size	Processes (ms)	Threads (ms)
250	0.156745	0.192143
500	0.164597	0.09477
750	0.181872	0.10057
1000	0.217576	0.215104
1250	0.237349	0.06092
1500	0.287873	0.071195
1750	0.30355	0.096401
2000	0.399755	0.087111
2250	0.415108	0.101222
2500	0.423935	0.110041
2750	0.455759	0.12482
3000	0.514027	0.163804
3250	0.540321	0.173559
3500	0.574699	0.18454
3750	0.588518	0.179966
4000	0.639635	0.18333
4250	0.695506	0.230694
4500	0.647994	0.238458
4750	0.735574	0.318693
5000	0.781476	0.243112
5250	0.71555	0.250199
5500	0.799848	0.314807
5750	0.927847	0.288706
6000	0.779362	0.279536
6250	0.981838	0.304615
6500	1.001875	0.343592
6750	0.979614	0.373457
7000	1.131009	0.352385
7250	1.143972	0.389231
7500	1.170123	0.419415
7750	1.075232	0.415141
8000	1.322372	0.402985
8250	1.32048	0.406503
8500	1.356371	0.41865
8750	1.625968	0.442178
9000	1.509625	0.419289
9250	1.487694	0.430994
9500	1.573882	0.451746
9750	1.705971	0.452698
10000	1.59198	0.466796

Test B

The graph below shows the trade-off point for when using multiple threads to search an array of a certain size is faster than searching through an array of 250 elements with one process. The graph is a bit misleading: it does not indicate that the search time for one process is invariant of array size. It is the time for specifically 250 elements, which was found to be approximately 106 microseconds. It took an array of 2250 elements for the multithreaded algorithm to become slower than the multiprocess algorithm, indicating that threads perform far faster than processes.

Figure 2: Results from Test B



Data:

Size	Search time (ms)
250	0.020869
500	0.02008
750	0.023637
1000	0.033189
1250	0.045138
1500	0.058647
1750	0.073827
2000	0.086896
2250	0.098077
2500	0.119823
2750	0.121146
3000	0.135383
3250	0.153539
3500	0.162763
3750	0.182762
4000	0.194125
4250	0.208528
4500	0.220368
4750	0.233679
5000	0.262649

Test C

We were surprised to see that, when the search is split up over more threads or more processes, the search always took more time. We believe that this is because, for such small arrays, the time taken to create a new process or a new thread, and then wait on all processes or threads to complete takes significantly longer than it would take to just perform a linear search through the entire array at once. Unfortunately, since the number of elements each process or thread can have cannot be bigger than 250, searching through a sufficiently large array would require far too many processes or threads.

Best fit equation for processes: $t = (3.28018 \text{ ms})n^{-0.66358}$

Best fit equation for threads: $t = (2.38345 \text{ ms})n^{-0.96293}$

Figure 3: Process results from Test C

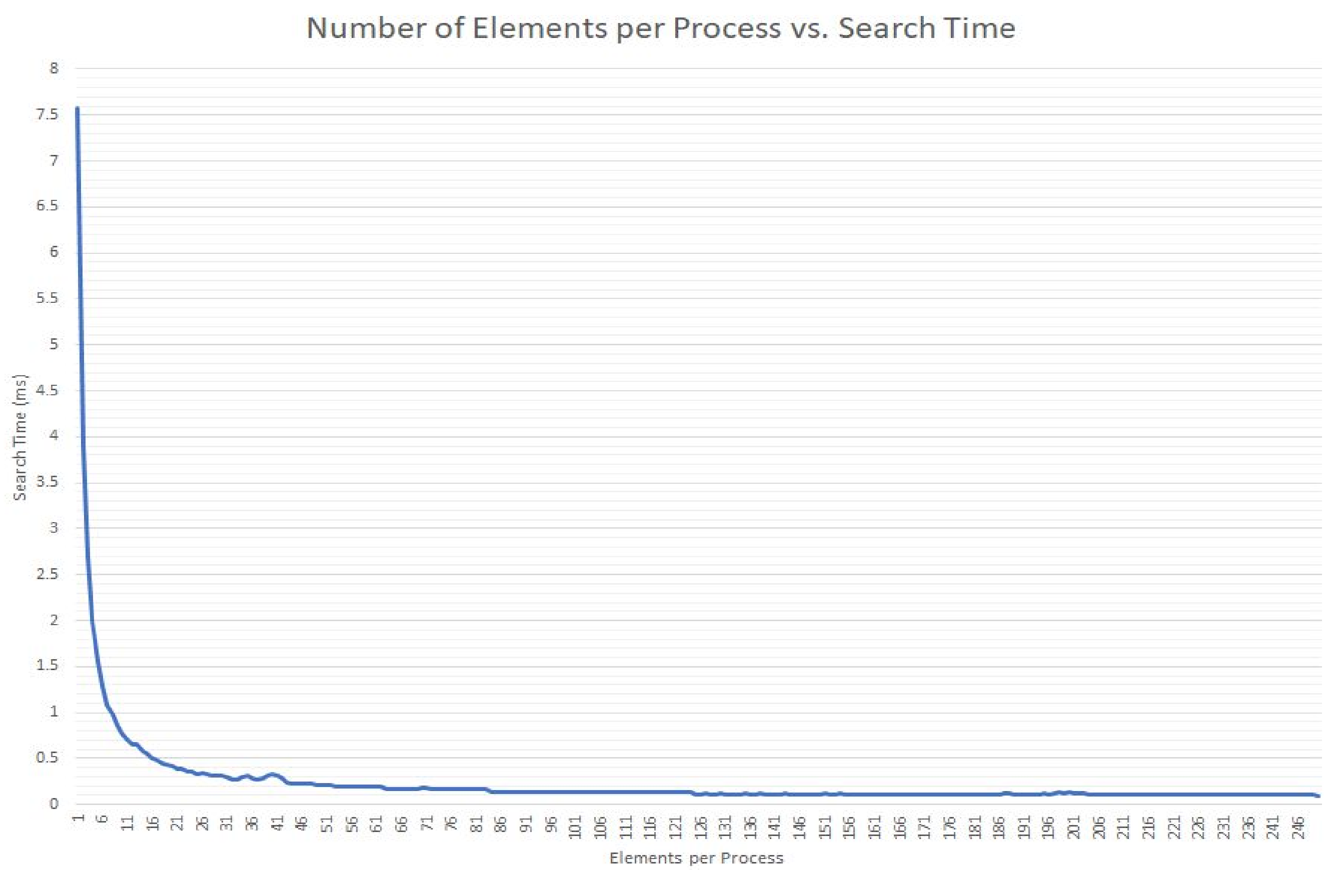
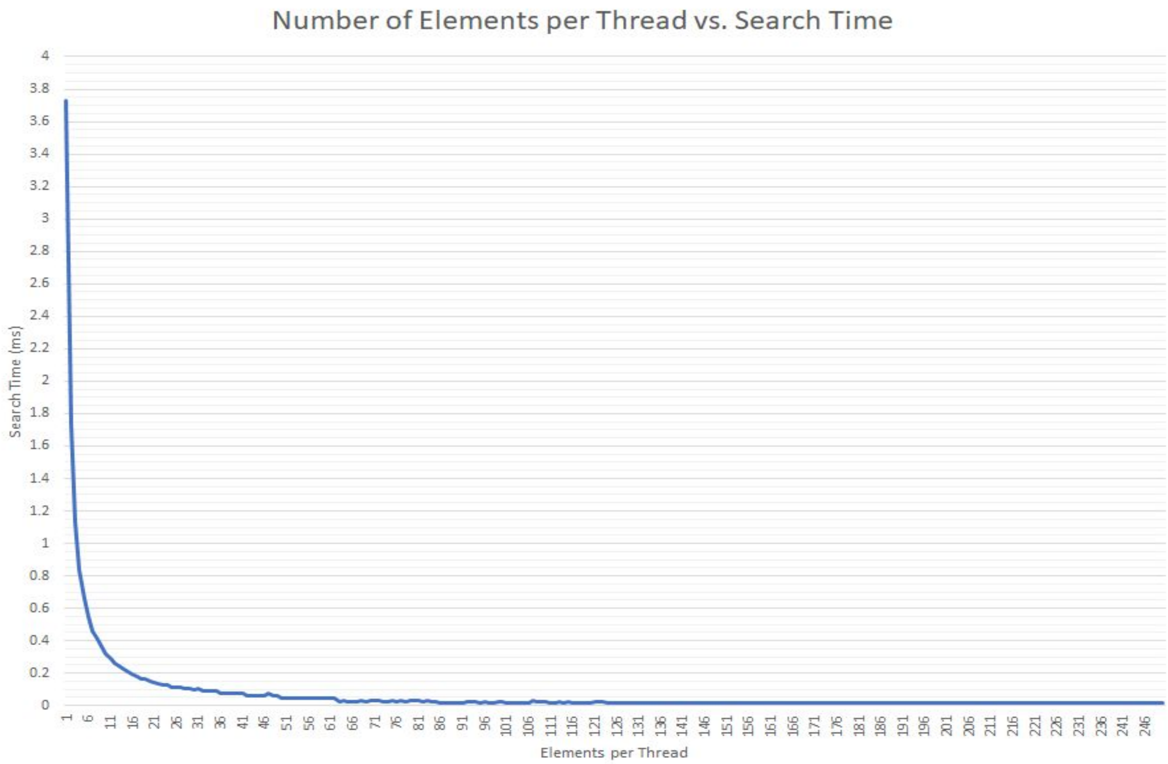


Figure 4: Thread results from Test C



Data (elements per process/thread, time for processes (ms), time for threads (ms)):

1	3.726819	7.582934
2	1.729758	3.890419
3	1.132517	2.690931
4	0.834505	2.002778
5	0.662785	1.550306
6	0.546976	1.288344
7	0.462431	1.082704
8	0.409558	0.986663
9	0.363886	0.857329
10	0.318924	0.765864
11	0.290888	0.700941
12	0.263949	0.649436
13	0.249297	0.64734
14	0.225696	0.582029
15	0.21018	0.549648
16	0.196234	0.504814
17	0.179913	0.477313
18	0.167375	0.441656
19	0.166436	0.435956
20	0.148385	0.410885
21	0.140507	0.387284
22	0.133316	0.384453
23	0.127325	0.362121
24	0.125173	0.359658
25	0.113982	0.331947
26	0.116318	0.340346
27	0.113946	0.334133
28	0.102654	0.310861
29	0.102594	0.316156
30	0.101115	0.309445
31	0.107997	0.305281
32	0.089523	0.275171
33	0.090643	0.273151
34	0.091123	0.301846
35	0.091837	0.320626
36	0.077443	0.288791
37	0.077476	0.277191
38	0.077418	0.282928
39	0.077799	0.309466
40	0.07915	0.324991
41	0.077566	0.319815
42	0.059618	0.288728
43	0.060482	0.244666
44	0.061625	0.22156
45	0.059404	0.220375
46	0.062214	0.220605
47	0.073451	0.229
48	0.061805	0.220891
49	0.06	0.216446
50	0.046214	0.205917
51	0.045191	0.204005
52	0.044991	0.208407
53	0.046763	0.189537
54	0.046235	0.192525
55	0.046248	0.188982
56	0.047213	0.198976
57	0.044458	0.195402
58	0.044476	0.19194
59	0.047286	0.199091
60	0.04691	0.190883
61	0.046787	0.195288
62	0.048336	0.202071
63	0.026838	0.171308
64	0.02777	0.173228
65	0.027392	0.163899
66	0.02668	0.170301
67	0.027436	0.166054
68	0.028064	0.170514
69	0.025848	0.170449
70	0.028491	0.173908
71	0.027765	0.175641
72	0.028403	0.161397
73	0.027291	0.166369
74	0.027213	0.168081
75	0.028909	0.165948
76	0.026191	0.161929
77	0.027812	0.166448
78	0.0272	0.164749
79	0.029777	0.171603
80	0.031016	0.162895
81	0.027826	0.166494

82	0.02665	0.161276
83	0.027727	0.161711
84	0.021382	0.142421
85	0.02029	0.136289
86	0.020232	0.140628
87	0.019908	0.137532
88	0.020201	0.140944
89	0.019093	0.133698
90	0.020037	0.138512
91	0.018775	0.137296
92	0.020596	0.135822
93	0.025986	0.140057
94	0.021066	0.135496
95	0.019328	0.14055
96	0.021671	0.133207
97	0.019435	0.138983
98	0.019872	0.133414
99	0.020761	0.139121
100	0.021792	0.132154
101	0.019099	0.134473
102	0.020028	0.136525
103	0.019006	0.132358
104	0.019735	0.138426
105	0.019685	0.132413
106	0.019733	0.138514
107	0.032076	0.135063
108	0.022436	0.134584
109	0.020499	0.132004
110	0.020628	0.13499
111	0.018271	0.133934
112	0.019704	0.132986
113	0.020283	0.136545
114	0.020018	0.130031
115	0.020892	0.134926
116	0.019162	0.131356
117	0.018708	0.135466
118	0.0195	0.13173
119	0.020028	0.129852
120	0.0185	0.135417
121	0.021977	0.130077
122	0.023886	0.137226
123	0.021599	0.130321
124	0.019821	0.134113
125	0.018243	0.113765
126	0.019522	0.112969
127	0.015964	0.116358
128	0.017759	0.111219
129	0.017755	0.111726
130	0.015755	0.11709
131	0.019186	0.111522
132	0.017237	0.113329
133	0.018219	0.114131
134	0.017734	0.112221
135	0.016703	0.11511
136	0.018168	0.112683
137	0.015977	0.110281
138	0.017194	0.115192
139	0.018392	0.111413
140	0.016231	0.113254
141	0.016858	0.112556
142	0.019314	0.110238
143	0.018029	0.115736
144	0.017946	0.111643
145	0.020132	0.112304
146	0.017437	0.113624
147	0.01668	0.111845
148	0.017661	0.113772
149	0.017408	0.111249
150	0.017796	0.11093
151	0.016208	0.114732
152	0.015793	0.111326
153	0.019767	0.109064
154	0.017434	0.114709
155	0.018794	0.110323
156	0.018774	0.109991
157	0.017047	0.112719
158	0.01767	0.109754
159	0.016817	0.112756
160	0.016396	0.11063
161	0.01854	0.11185
162	0.018025	0.112111
163	0.017945	0.109585
164	0.018613	0.111647
165	0.017562	0.113675
166	0.018113	0.108993
167	0.016581	0.110693
168	0.01779	0.110415
169	0.017408	0.107636
170	0.017488	0.111284
171	0.016754	0.108316
172	0.018071	0.107372
173	0.016495	0.110621
174	0.018745	0.106352
175	0.017895	0.107983
176	0.019145	0.111383
177	0.016335	0.106591
178	0.018454	0.106011
179	0.017186	0.108903
180	0.018308	0.10911
181	0.01699	0.111598
182	0.017444	0.108481
183	0.019022	0.107039
184	0.016508	0.114005
185	0.016551	0.111682
186	0.016043	0.110505
187	0.018201	0.11835
188	0.016361	0.114959
189	0.018389	0.109354
190	0.018673	0.114057
191	0.01633	0.109637
192	0.016433	0.11214
193	0.018095	0.112192
194	0.018617	0.107771
195	0.017288	0.114657
196	0.017059	0.110402
197	0.017582	0.127672
198	0.017978	0.13114
199	0.016854	0.124693
200	0.019457	0.133913
201	0.015903	0.128852
202	0.018607	0.128213
203	0.017164	0.121459
204	0.016104	0.105282
205	0.019271	0.107057
206	0.019391	0.106234
207	0.01866	0.104755
208	0.016695	0.110116
209	0.016432	0.105197
210	0.01727	0.105874
211	0.018098	0.110455
212	0.019335	0.104569
213	0.016569	0.105626
214	0.017214	0.106704
215	0.015957	0.106282
216	0.016517	0.106207
217	0.015667	0.106093
218	0.0169	0.105726
219	0.016725	0.108821
220	0.01585	0.105556
221	0.016411	0.10685
222	0.015591	0.107196
223	0.018142	0.105481
224	0.016336	0.104805
225	0.01611	0.107877
226	0.016753	0.105015
227	0.015707	0.104505
228	0.019575	0.106999
229	0.017272	0.103469
230	0.017823	0.104936
231	0.017901	0.105374
232	0.017401	0.105154
233	0.017612	0.108744
234	0.018435	0.105498
235	0.017769	0.104802
236	0.017408	0.10643
237	0.017355	0.104465
238	0.016824	0.102617
239	0.016948	0.105111
240	0.018295	0.103265
241	0.017399	0.104894
242	0.01805	0.107364
243	0.015361	0.102969
244	0.01698	0.102339
245	0.016591	0.106443
246	0.017558	0.100433
247	0.017292	0.100802
248	0.016737	0.105307
249	0.015685	0.099818
250	0.013886	0.092381