

Asst2: Spooky Searching
results.pdf

Students: Aditya Oka (aao96, 01), Chris DeAngelis (cjd237)

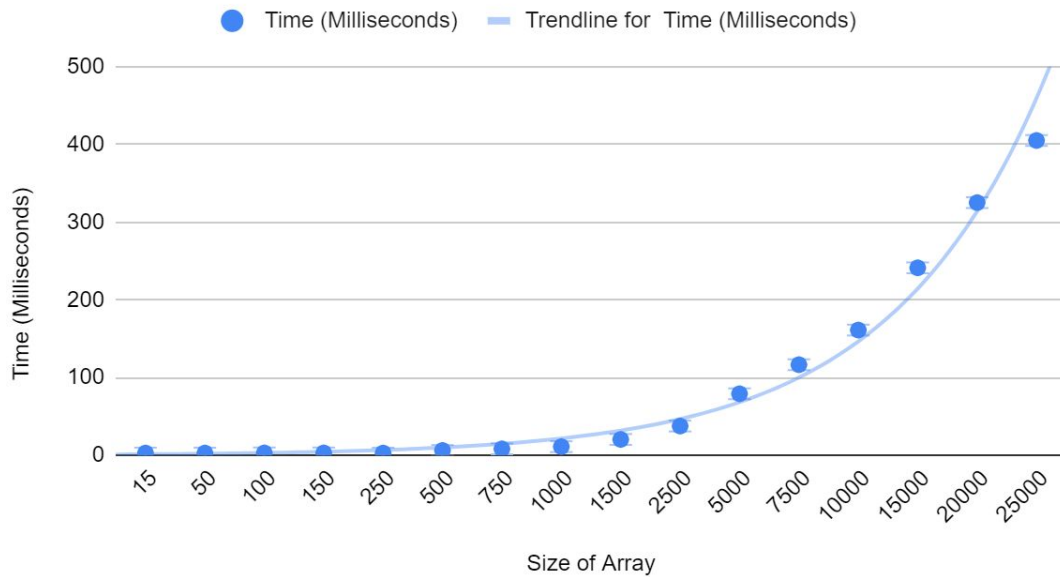
Q1 Results

As the size of the array continues to increase, the time to complete gets longer gradually, as expected. However, as the size increases, the time it takes for threads to complete gets more efficient than processes.

1: Multi-threaded

| Size of Array | Time (Milliseconds) |
|---------------|---------------------|
| 15 | 3.06 |
| 50 | 3.08 |
| 100 | 3.21667 |
| 150 | 3.2 |
| 250 | 3.06 |
| 500 | 6.4 |
| 750 | 8.44 |
| 1000 | 11.28 |
| 1500 | 20.58 |
| 2500 | 37.78 |
| 5000 | 79.26 |
| 7500 | 116.64 |
| 10000 | 161.16 |
| 15000 | 241.18 |
| 20000 | 324.98 |
| 25000 | 404.88 |

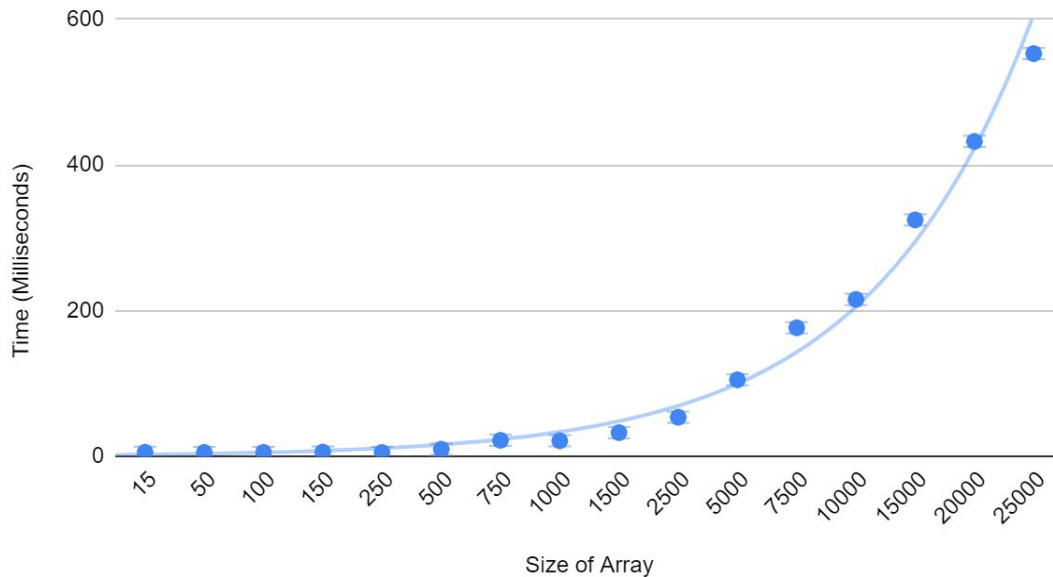
Time (Milliseconds) vs. Size of Array



2: Multi-processed

| Size of Array | Time (Milliseconds) |
|---------------|---------------------|
| 15 | 5.88 |
| 50 | 5.64 |
| 100 | 5.68 |
| 150 | 6.12 |
| 250 | 5.62 |
| 500 | 10.1 |
| 750 | 22.28 |
| 1000 | 21.64 |
| 1500 | 32.78 |
| 2500 | 53.98 |
| 5000 | 105.35 |
| 7500 | 176.72 |
| 10000 | 215.66 |
| 15000 | 324.84 |
| 20000 | 432.46 |
| 25000 | 553.04 |

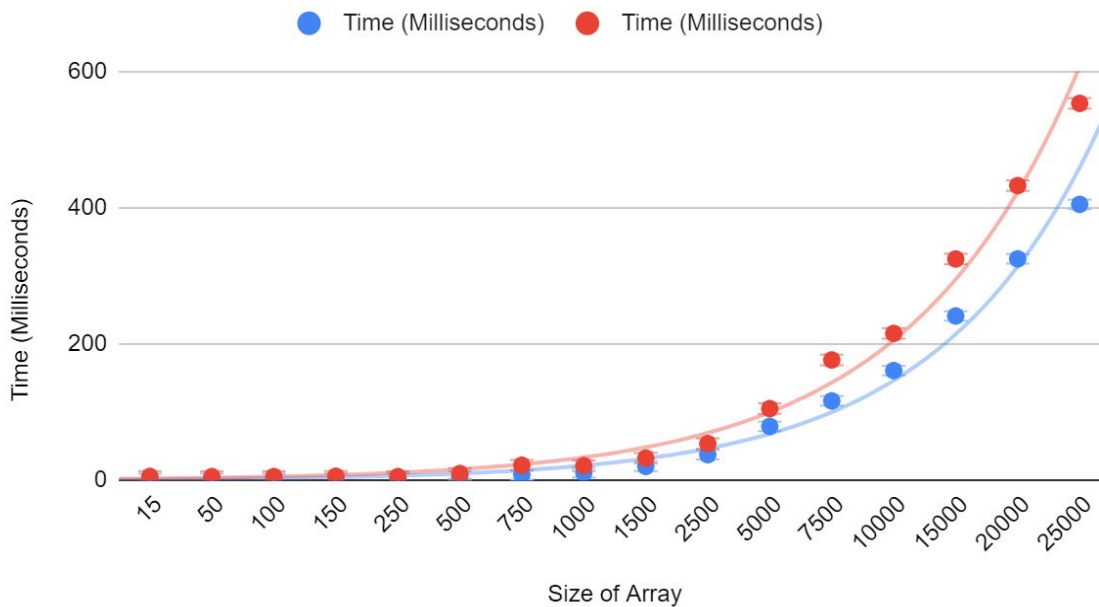
Time (Milliseconds) vs. Size of Array



Q2 Results

The tradeoff point of performance for processes vs threads, when we are taking the time for processes to search an array of size 250, seems to be most similar to the time for threads to search an array of size 500. Looking at other tradeoff points of performance, we observed that the time for processes to search arrays of sizes 150, 500, 1000, 15000 matches the time for threads to search arrays of sizes 500, 1000, 1500, and 20000 respectively.

Time (Milliseconds) vs. Size of Array



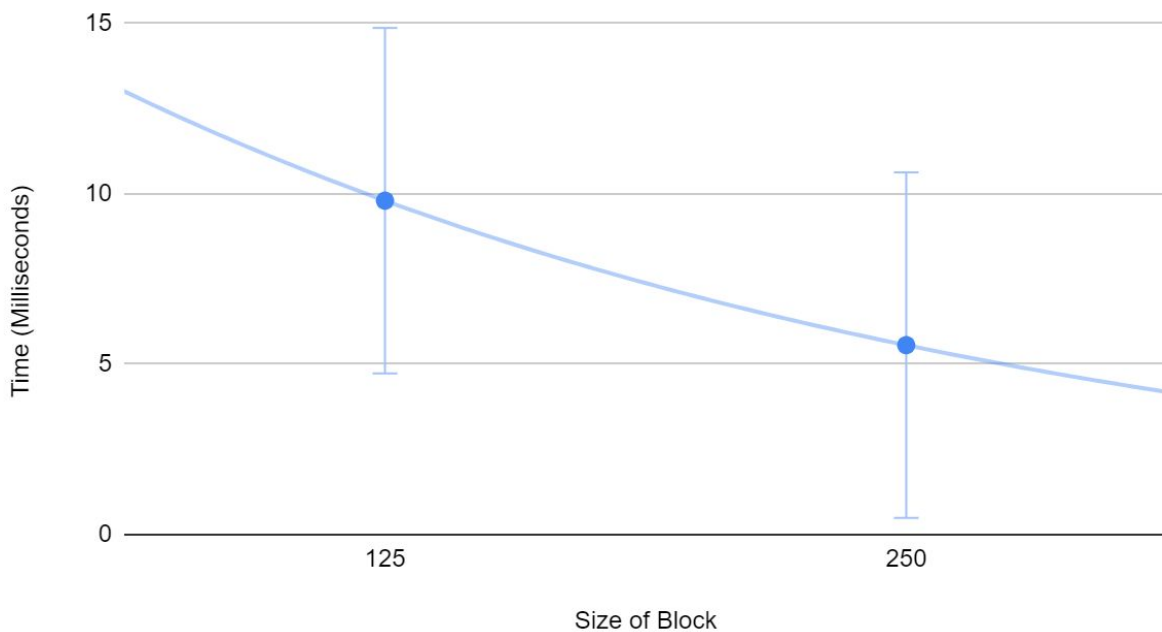
Q3 Results

For both multi-processed and multi-threaded searches, the more times the array is divided (i.e. the more threads/processes are created to search or the smaller the block) the longer it takes. This inefficiency starts building up as soon as even a single extra unnecessary thread/process is introduced. Therefore it is recommended to use only as many threads/processes as needed, based on this data.

1: Processes

| Size of Block | Time (Milliseconds) |
|---------------|---------------------|
| 125 | 9.8 |
| 250 | 5.56 |

Time (Milliseconds) vs. Size of Block



2: Threads

| Size of Block | Time (Millisecond) |
|---------------|--------------------|
| 25 | 36.42 |
| 50 | 15.68 |
| 125 | 5.48 |
| 250 | 3.12 |

Time (Millisecond) vs. Size of Block

