COMPSCI 2XB3 Assignment 1 Analysis

Frank Su, 001411435

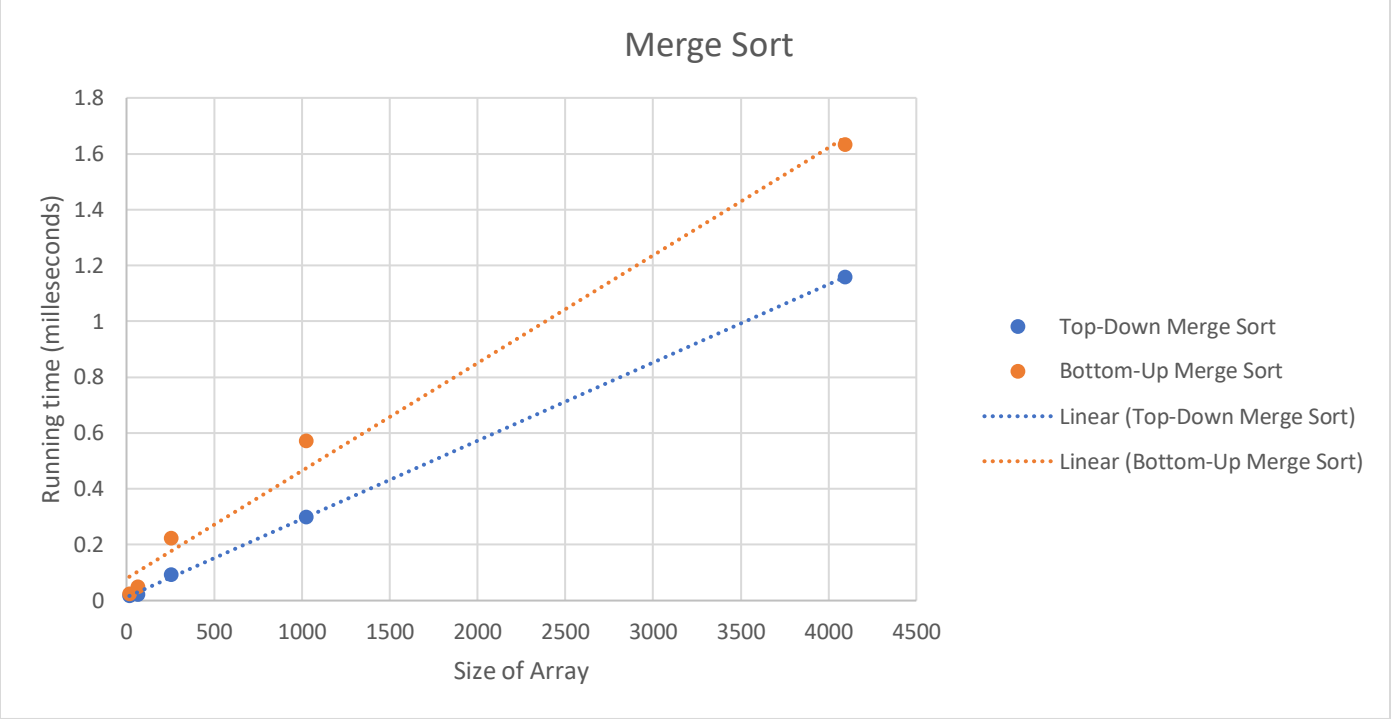
Table 1: Execution time of Insertion Sort, Merge Sort and Heap Sort

Size of Array	Execution Time (in milliseconds)							
	Basic Insertion Sort	Comparable Insertion Sort	Binary Insertion Sort	Top-Down Merge Sort	Bottom-Up Merge Sort	Heap Sort	Basic Quick Sort	Three Partition Quick Sort
16	0.05123	0.062577	0.02048	0.017067	0.02342	0.03252	0.0603	0.02
64	0.21049	0.131982	0.043236	0.022186	0.050062	0.08192	0.031289	0.156444
256	0.97337	0.856178	0.389689	0.075093	0.236658	0.325405	0.129707	0.513707
1024	6.40179	8.320002	4.106242	0.277618	0.59278	0.869832	0.496072	1.272036
4096	26.397589	90.94	52.712122	1.112178	1.632712	2.055396	1.780622	2.373974

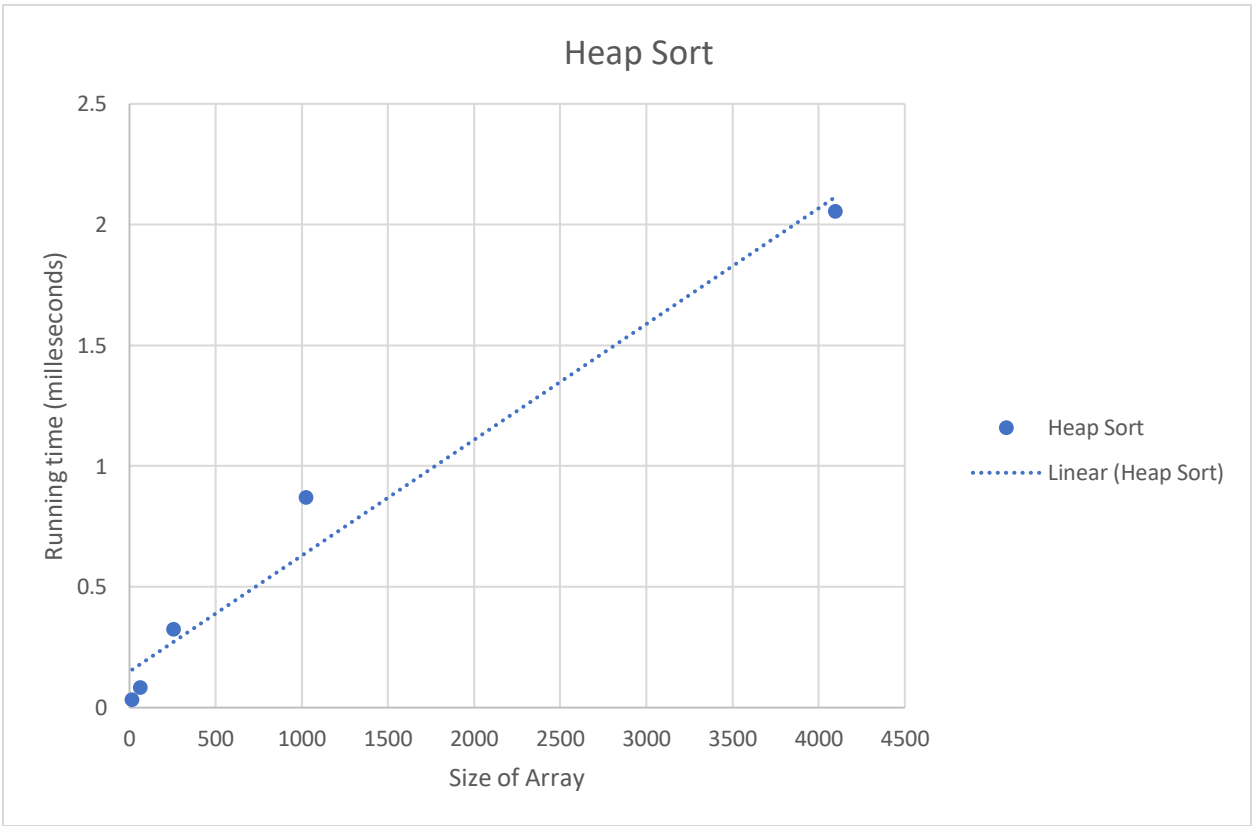
3.1.1 part 1: Results plotted using normal scale

Graph 1: Insertion Sort results

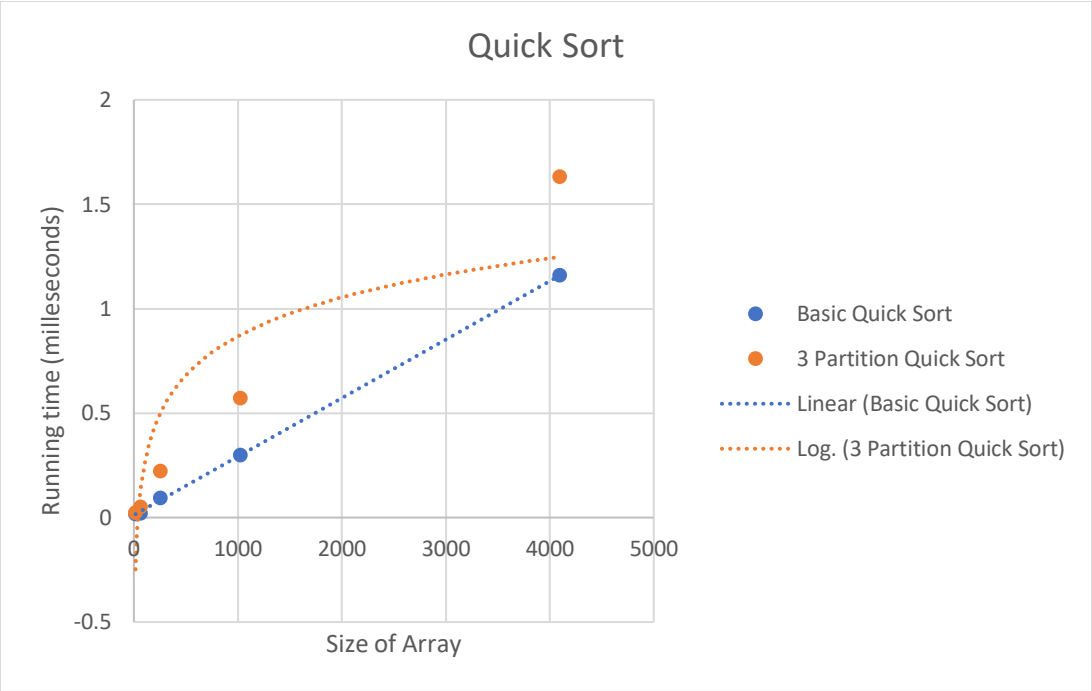
Graph 2: Merge Sort results



Graph 3: Heap Sort results

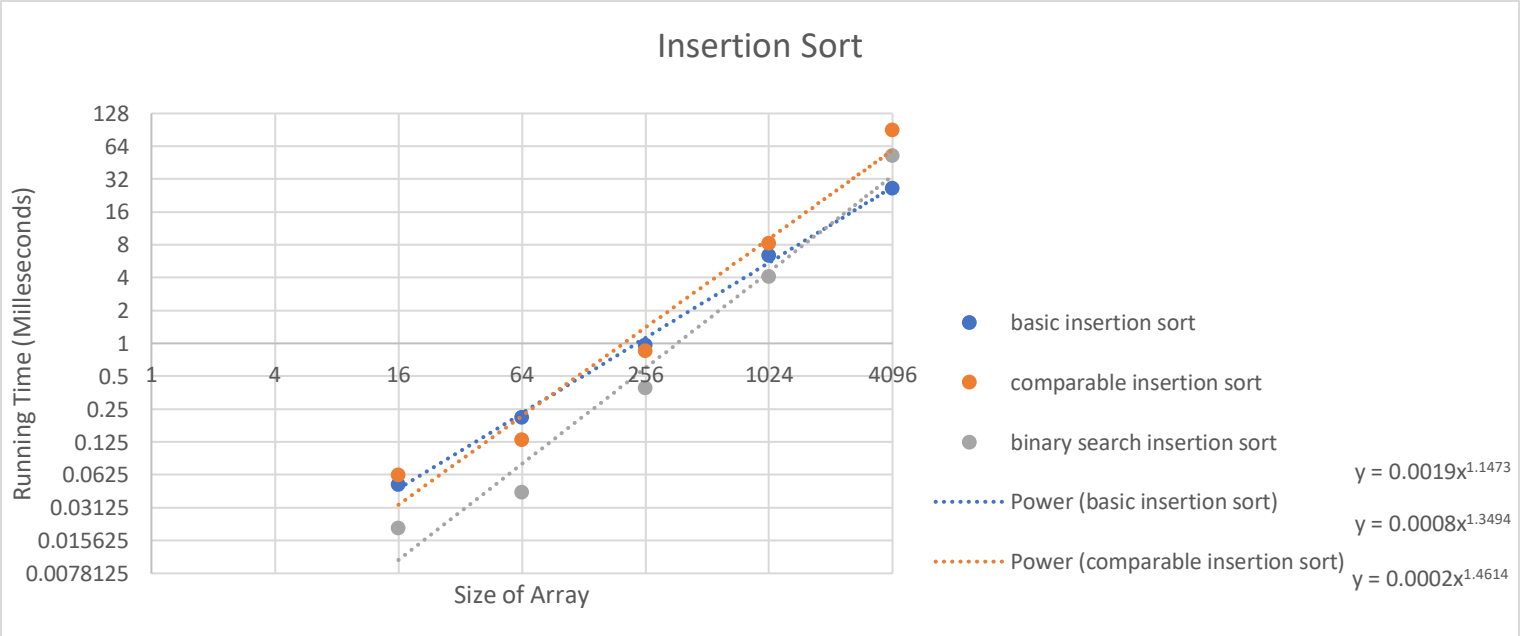


Graph 4: Quick Sort Results

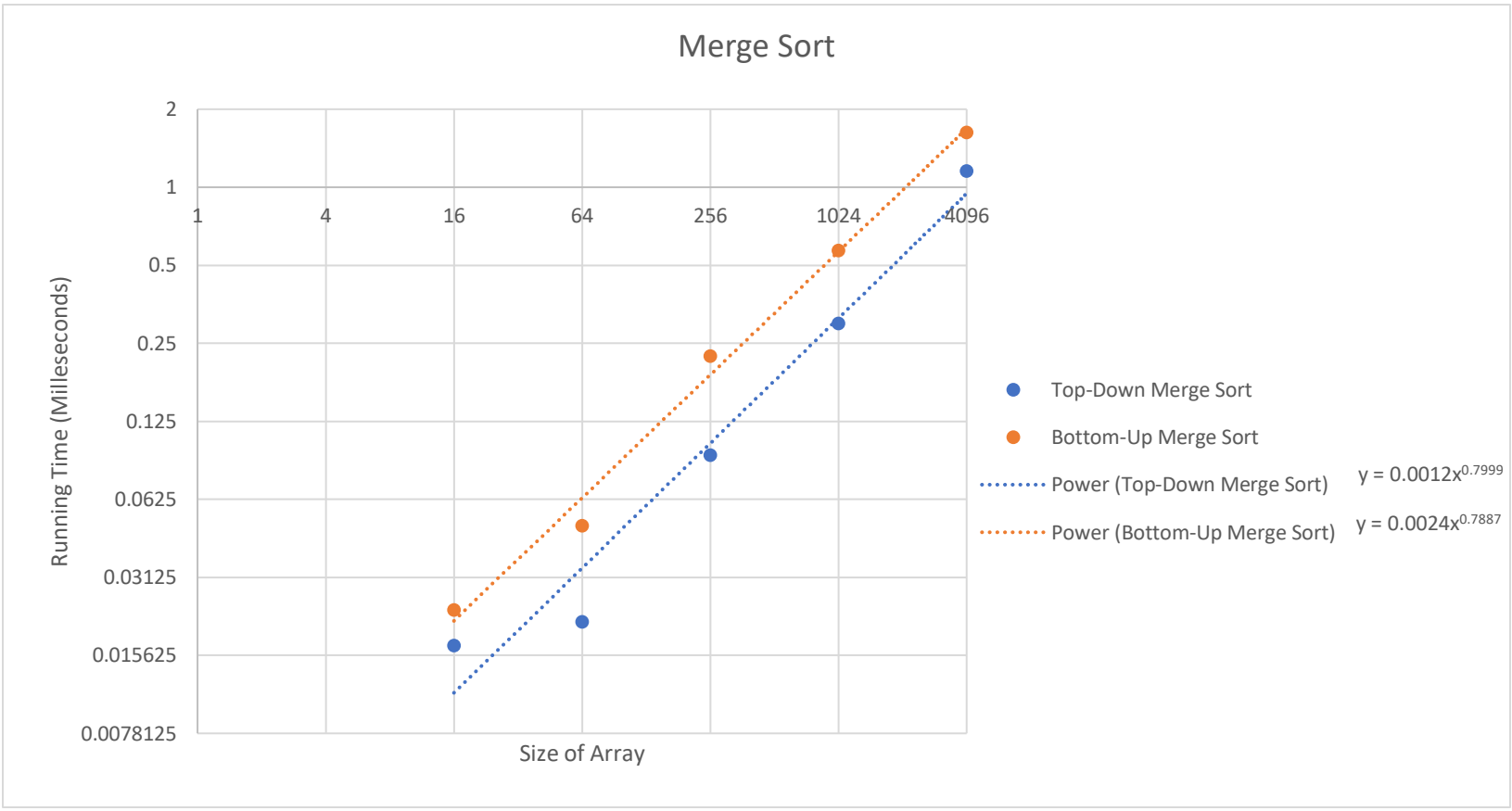


3.1.1 Part 2: Results plotted using log-log scale:

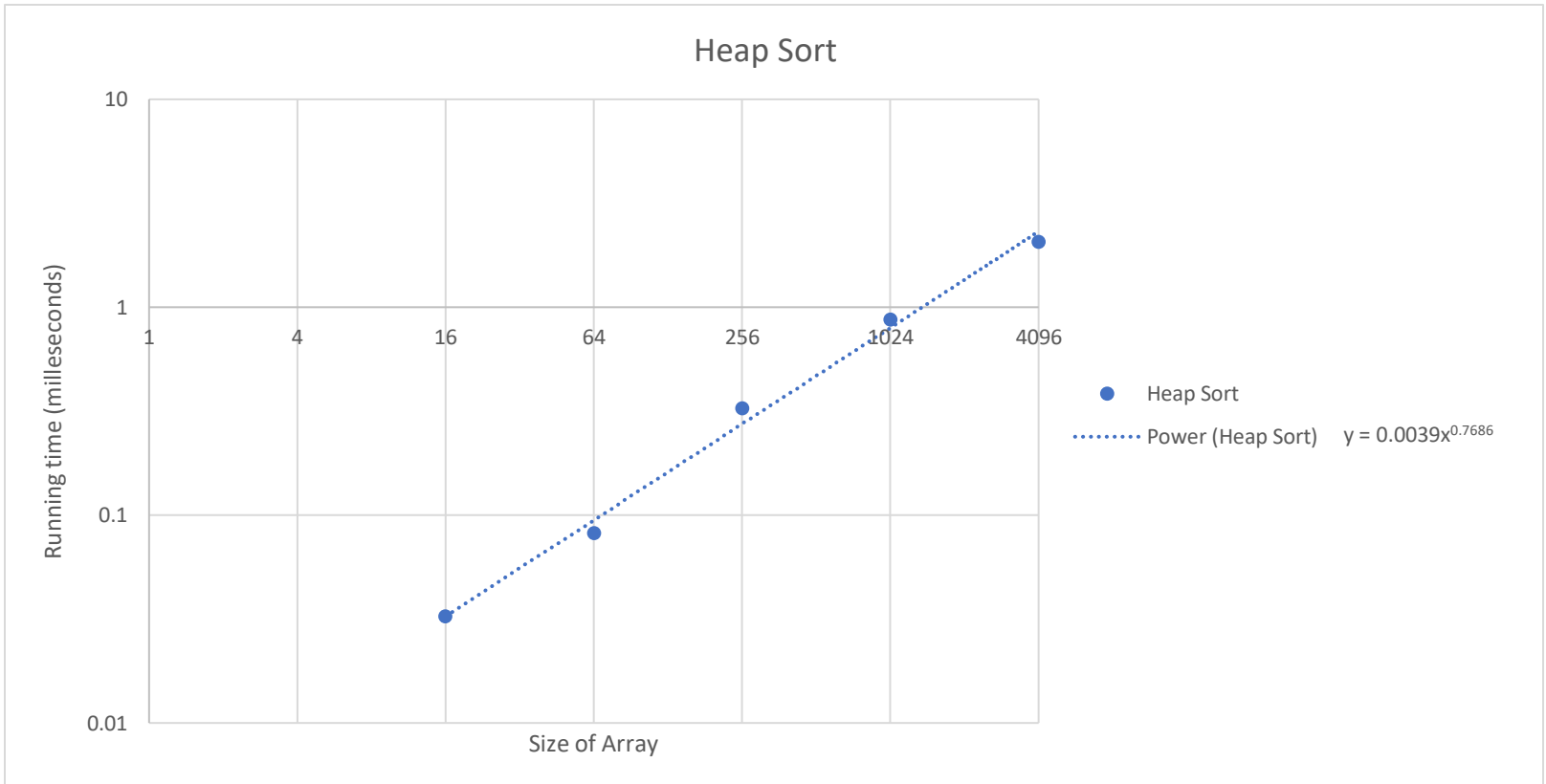
Graph 1: Insertion sort results



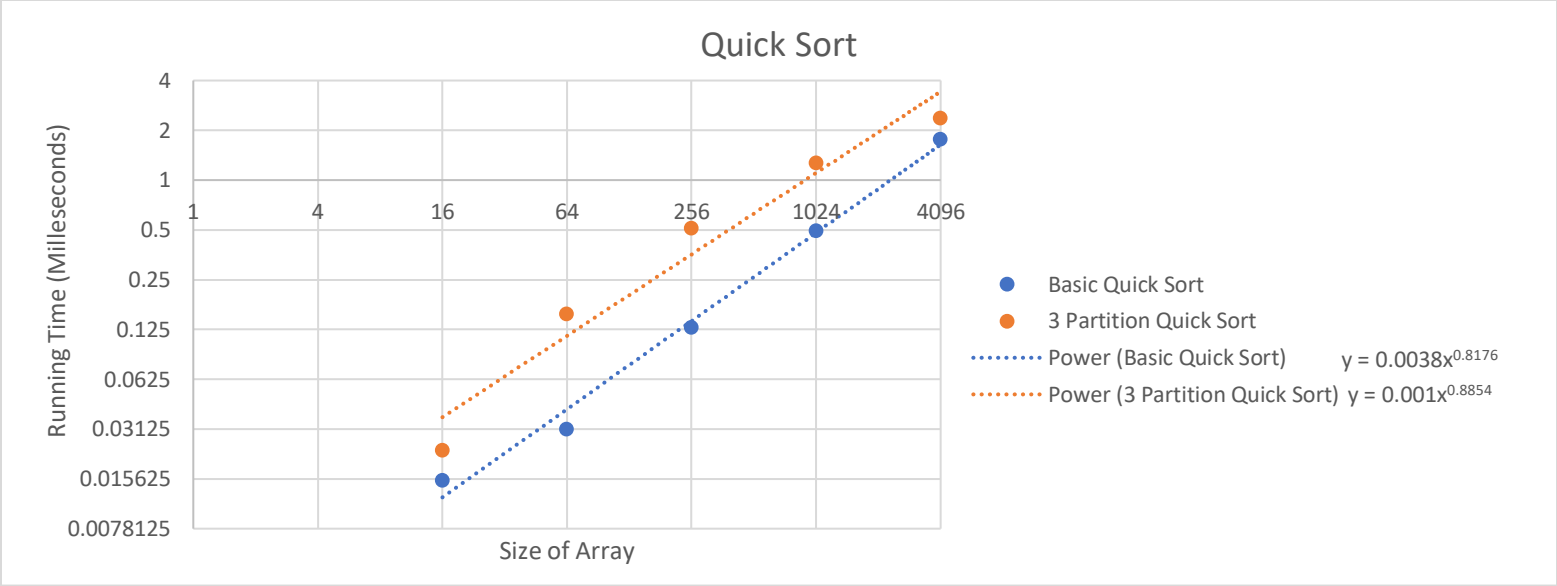
Graph 2: Merge Sort results



Graph 3: Heap Sort results



Graph 4: Quick Sort Results



3.1.2 Hypothesis of Running time

To form a hypothesis of running time for each of the algorithms implemented, we can use the power equation obtained by fitting a straight line through the data points of the log-log scale graphs. This equation serves as an estimate that we can use to calculate the running time for arrays of larger size. The hypothesis for the algorithms are thus as follows:

Insertion sort:

- Basic insertion sort: The running time is about $0.0019N^{1.1473}$ ms
- Comparable Insertion sort: The running time is about $0.0008N^{0.1.3494}$ ms
- Binary search insertion sort: The running time is about $0.0002N^{1.4614}$ ms

Merge sort:

- Top-Down Merge Sort: The running time is about $0.0012N^{0.7999}$ ms
- Bottom-Up Merge Sort: The running time is about $0.0024N^{0.7887}$ ms

Heap sort:

- The running time is about $0.0039N^{0.7686}$ ms

Quick sort:

- Basic quick sort: The running time is about $0.0038N^{0.8176}$ ms
- 3 Partition quick sort: The running time is about $0.001N^{0.8854}$ ms

3.1.3 Use hypotheses to predict running time of arrays of size 2^{14} and 2^{16}

Using the equations formulated in 3.1.2 as our hypotheses, the estimated running times for arrays of size 2^{14} and 2^{16} are as follows:

Insertion sort:

- Basic insertion sort:
  - Array of size 2^{14}: The running time should be approximately $0.0019(16384)^{1.1473}$ = 130 ms
  - Array of size 2^{16}: The running time should be approximately $0.0019(65536)^{1.1473}$ = 637.82ms
- Comparable insertion sort:
  - Array of size 2^{14}: The running time should be approximately $0.0008(16384)^{1.3494}$ = 389 ms
  - Array of size 2^{16}: The running time should be approximately $0.0008(65536)^{1.3494}$ = 2526 ms
- Binary search insertion sort:
  - Array of size 2^{14}: The running time should be approximately $0.0002(16384)^{1.4614}$ = 288 ms
  - Array of size 2^{16}: The running time should be approximately $0.0002(65536)^{1.4614}$ = 2186 ms

Merge sort:

- Top-Down merge sort:
  - Array of size 2^{14}: The running time should be approximately $0.0012(16384)^{0.799}$ = 2.796 ms
  - Array of size 2^{16}: The running time should be approximately $0.0012(65536)^{0.799}$ = 8.46
- Bottom-up merge sort:
  - Array of size 2^{14}: The running time should be approximately $0.0024(16384)^{0.7887}$ = 5.059 ms
  - Array of size 2^{16}: The running time should be approximately $0.0024(65536)^{0.7887}$ = 15.09 ms

Heap sort:

- Array of size 2^{14}: The running time should be approximately $0.0039(16384)^{0.7686}$ = 6.76 ms
- Array of size 2^{16}: The running time should be approximately $0.0039(65536)^{0.7686}$ = 19.634

Quick sort:

- Basic quick sort:
  - Array of size 2^{14}: The running time should be approximately $0.0038(16384)^{0.8176}$ = 10.60 ms
  - Array of size 2^{16}: The running time should be approximately $0.0038(65536)^{0.8176}$ = 32.94 ms
- 3 Partition quick sort:
  - Array of size 2^{14}: The running time should be approximately $0.001(16384)^{0.8854}$ = 5.388 ms

- Array of size 2^{16}: The running time should be approximately $0.001(65536)^{0.8854}$ = 18.9 ms

3.1.4 Verify predictions by running program

When running my program for array sizes of 2^{14} and 2^{16}, the results are as follows:

Insertion sort:

- Basic insertion sort:
  - Array of size 2^{14}: 415 ms
  - Array of size 2^{16}: 13413 ms
- Comparable insertion sort:
  - Array of size 2^{14}: 1385 ms
  - Array of size 2^{16}: 35375 ms
- Binary search insertion sort:
  - Array of size 2^{14}: 670 ms
  - Array of size 2^{16}: 11997 ms

When comparing these results with the predicted results, it is easy to see that the actual results are much higher than predicted. One reason can be attributed to the fact that the prediction doesn't take into account an average time across multiple trials, as the running time slightly varies across different trials. In addition, the array sizes for the initial observations may not have been big enough to model insertion sorts running time, as unlike selection sort that has a $O(N^2)$ running time for its worst average and best case, insertion sorts best case can be linear. The important factor is to compare the running time of insertion sort with the nlogn sorting methods.

Merge sort:

- Top-Down merge sort:
  - Array of size 2^{14}: 5.43 ms
  - Array of size 2^{16}: 30.98 ms
- Bottom-up merge sort:
  - Array of size 2^{14}: 11 ms
  - Array of size 2^{16}: 60 ms

Comparing these results with the insertion sort results, we can see that although the running times are still higher than predicted, they are much more accurate. In addition, we can see the stark difference between the running times of a O(N^2) insertion sort vs merge sort, which is a O(nlogn) sort.

Heap sort:

- Array of size 2^{14}: 22.19 ms
- Array of size 2^{16}: 56 ms

Similarly to merge sort, although the running times are a bit higher than predicted, they still model that of a O(nlogn) sorting method, and runs much quicker than insertion sort.

Quick sort:

- Basic quick sort:
  - Array of size 2^{14}: 6 ms
  - Array of size 2^{16}: 40.99 ms
- 3 Partition quick sort:
  - Array of size 2^{14}: 10 ms
  - Array of size 2^{16}: 30 ms

For Quicksort, the predicted running times are fairly close to the actual running time, confirming the hypothesized model.