Program Structures and Algorithms
Fall 2023

NAME: Abhishek Udayakumar Guragol
NUID: 002765039

Task:

In This assignment the main task was:

Determine--for sorting algorithms--what is the best predictor of total execution time: comparisons, swaps/copies, hits (array accesses)

Relationship Conclusion:

In order to draw a conclusion we must first find the correlations between Compares, Swaps and Hits for all the sorting algorithms, i.e how the execution time changes with each of these values.

Below shows the table that shows the sort and the which parameter has the highest influence on the total execution time:

Sl No	Algorithm	Highest Influencer
1	Merge Sort	Compares
2	Quick Sort (Dual - Pivot)	Hits
3	Heap Sort	Compares & Hits

Dual pivot quick sort outperforms merge sort, and merge sort outperforms heap sort. However, as the size grows, merge sort becomes comparable to dual pivot quick sort. Time is proportional to the number of compares, swaps, and hits. Although swap is a costly operation and the number of hits is higher, the number of compares and array size contribute more to the raw time taken.

Evidence to support that conclusion:

Observations:

1) Merge sort

Size	Time	Normalize time	Compares	Swaps	Hits	Log(size)
16000	3.58	2.99	206767	14017	434704	13.96578428
32000	8.67	3.14	445486	28027	937504	14.96578428
64000	16.58	2.97	955810	56077	1994805	15.96578428
128000	37.71	3.16	2043518	112173	4245685	16.96578428
256000	79.29	3.14	4,303,219	224334	9003208	17.96578428

2) (dual-pivot) quick sort

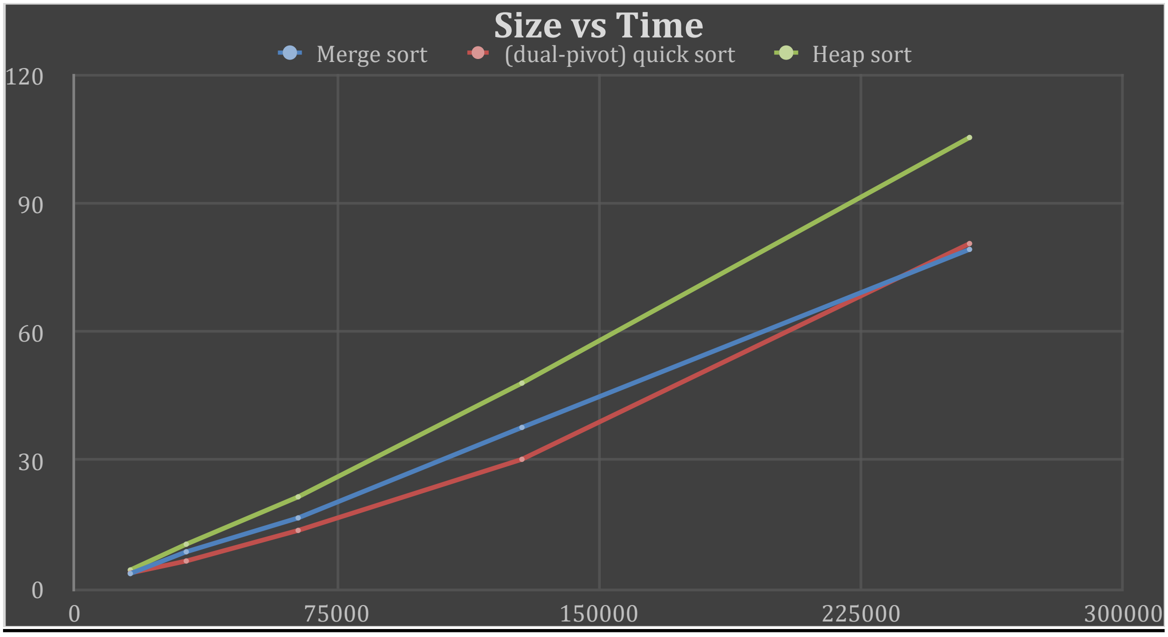
Size	**Time**	**Normalize time**	**Compares**	**Swaps**	**Hits**	**Log(size)**
16000	3.72	2.72	264,768	111,368	702,986	13.96578428
32000	6.52	2.51	574,025	240,782	1,522,039	14.96578428
64000	13.66	2.45	1,234,828	512,460	3,255,929	15.96578428
128000	30.27	2.54	2,649,857	1,115,510	7,051,684	16.96578428
256000	80.65	3.18	5,667,442	2,385,104	15,093,864	17.96578428

3) Heapsort

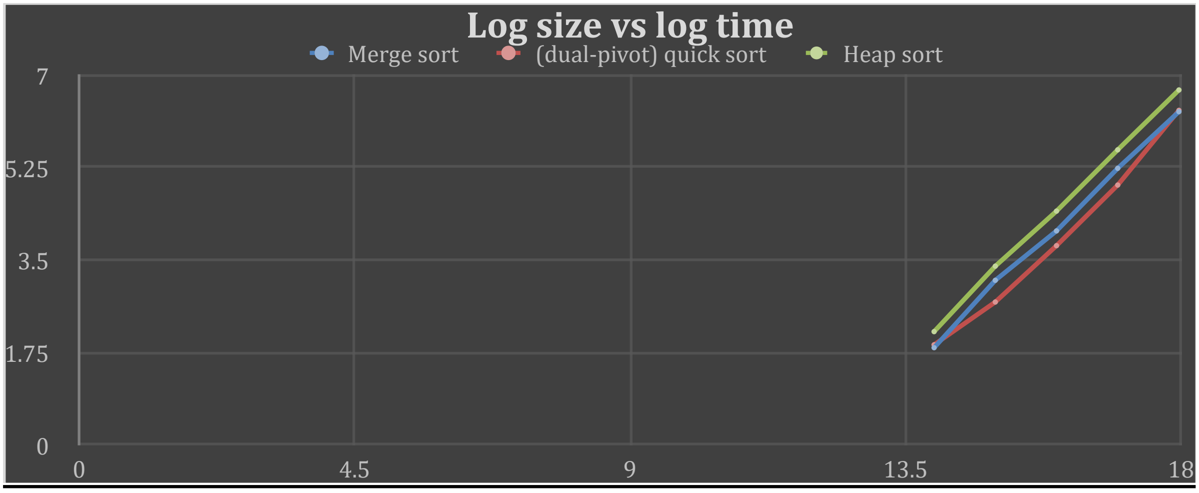
Size	**Time**	**Normalize time**	**Compares**	**Swaps**	**Hits**	**Log(size)**
16000	4.42	3.67	397,646	209,248	1,632,282	13.96578428
32000	10.44	4.02	859,312	450,508	3,520,656	14.96578428
64000	21.5	3.86	1,846,651	965,040	7,553,463	15.96578428
128000	48.06	4.03	3,949,237	2,057,957	16,130,303	16.96578428
256000	105.42	4.15	8,410,502	4,372,027	34,309,112	17.96578428

Graphical Representation:

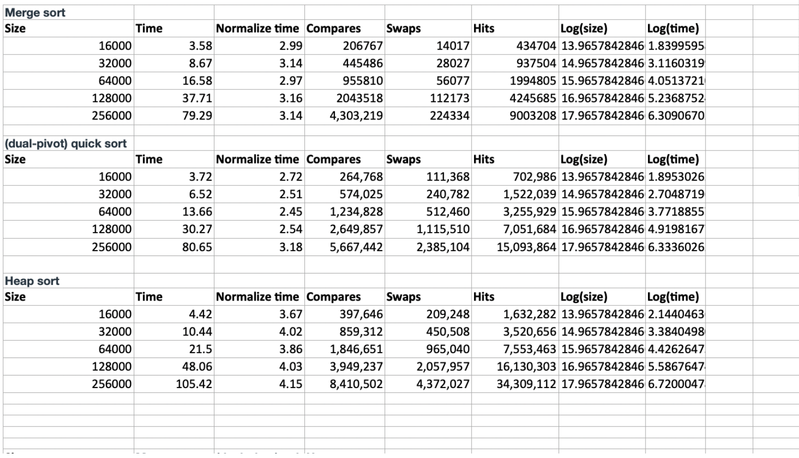
1. **Size VS Time**



2. Log(size) vs Log(time)



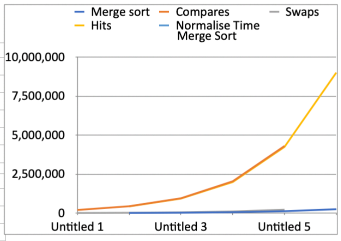
3. Correlations between the elements:

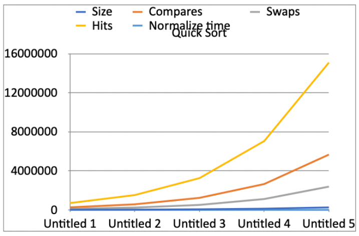


Merge sort							
Size	Time	Normalize time	Compares	Swaps	Hits	Log(size)	Log(time)
16000	3.58	2.99	206767	14017	434704	13.9657842846	1.8399595
32000	8.67	3.14	445486	28027	937504	14.9657842846	3.1160319
64000	16.58	2.97	955810	56077	1994805	15.9657842846	4.0513721
128000	37.71	3.16	2043518	112173	4245685	16.9657842846	5.2368752
256000	79.29	3.14	4,303,219	224334	9003208	17.9657842846	6.3090670

(dual-pivot) quick sort							
Size	Time	Normalize time	Compares	Swaps	Hits	Log(size)	Log(time)
16000	3.72	2.72	264,768	111,368	702,986	13.9657842846	1.8953026
32000	6.52	2.51	574,025	240,782	1,522,039	14.9657842846	2.7048719
64000	13.66	2.45	1,234,828	512,460	3,255,929	15.9657842846	3.7718855
128000	30.27	2.54	2,649,857	1,115,510	7,051,684	16.9657842846	4.9198167
256000	80.65	3.18	5,667,442	2,385,104	15,093,864	17.9657842846	6.3336026

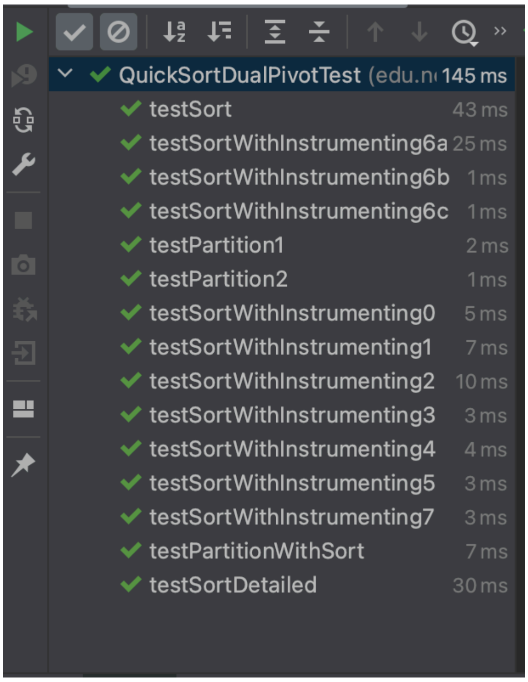
Heap sort							
Size	Time	Normalize time	Compares	Swaps	Hits	Log(size)	Log(time)
16000	4.42	3.67	397,646	209,248	1,632,282	13.9657842846	2.1440463
32000	10.44	4.02	859,312	450,508	3,520,656	14.9657842846	3.3840498
64000	21.5	3.86	1,846,651	965,040	7,553,463	15.9657842846	4.4262647
128000	48.06	4.03	3,949,237	2,057,957	16,130,303	16.9657842846	5.5867647
256000	105.42	4.15	8,410,502	4,372,027	34,309,112	17.9657842846	6.7200047



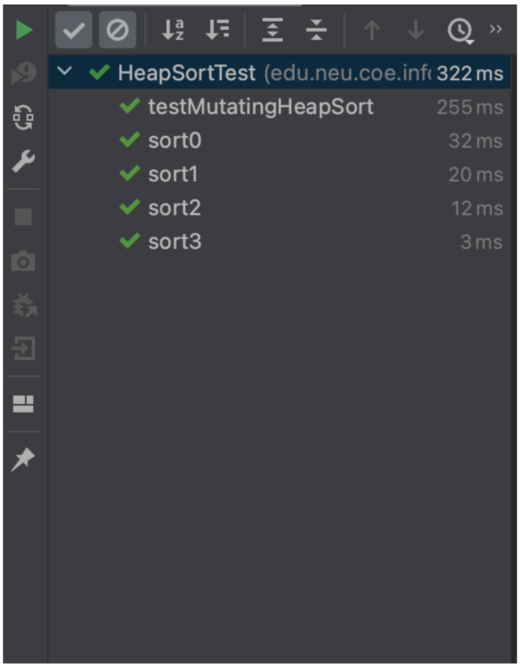


Screenshots of run and/or Unit Test:

Unit test

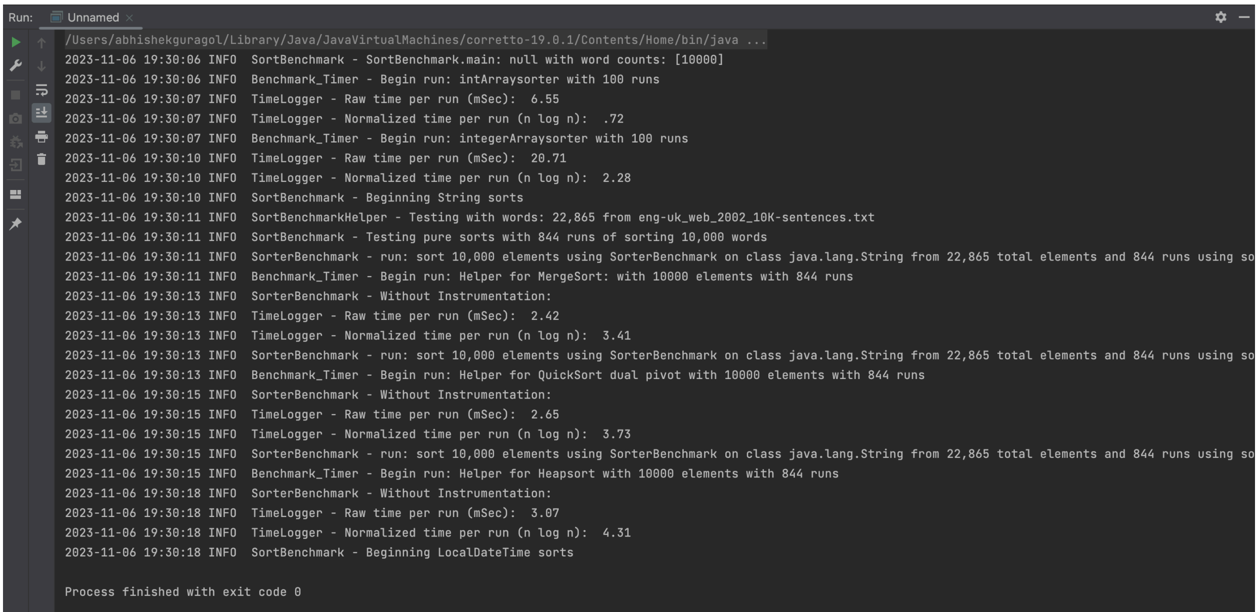


```
QuickSortDualPivotTest (edu.n  145 ms
  testSort                          43 ms
  testSortWithInstrumenting6a       25 ms
  testSortWithInstrumenting6b        1 ms
  testSortWithInstrumenting6c        1 ms
  testPartition1                     2 ms
  testPartition2                     1 ms
  testSortWithInstrumenting0         5 ms
  testSortWithInstrumenting1         7 ms
  testSortWithInstrumenting2        10 ms
  testSortWithInstrumenting3         3 ms
  testSortWithInstrumenting4         4 ms
  testSortWithInstrumenting5         3 ms
  testSortWithInstrumenting7         3 ms
  testPartitionWithSort              7 ms
  testSortDetailed                  30 ms
```



```
HeapSortTest (edu.neu.coe.inf  322 ms
  testMutatingHeapSort      255 ms
  sort0                      32 ms
  sort1                      20 ms
  sort2                      12 ms
  sort3                       3 ms
```

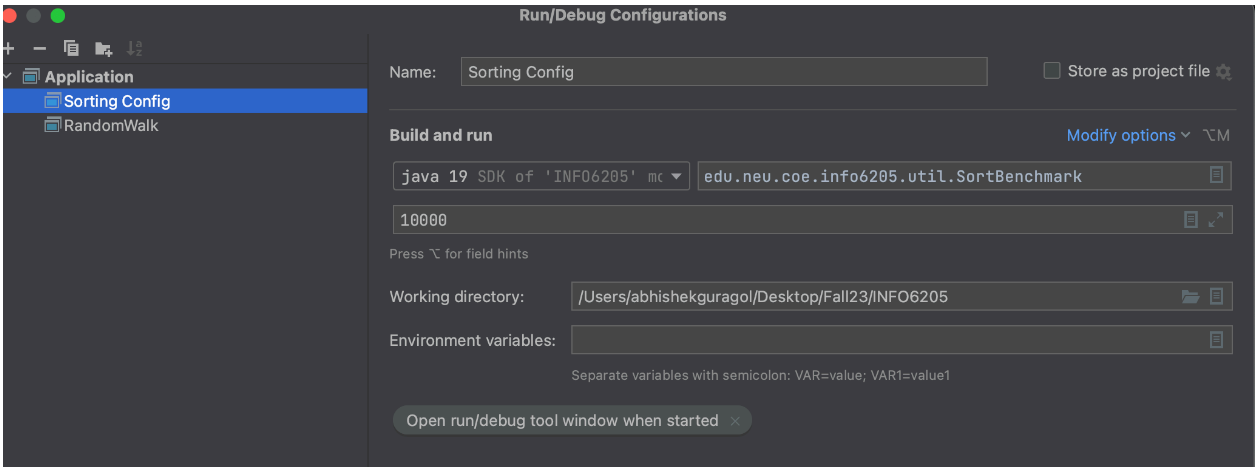
Results:



```
/Users/abhishekguragol/Library/Java/JavaVirtualMachines/corretto-19.0.1/Contents/Home/bin/java ...
2023-11-06 19:30:06 INFO  SortBenchmark - SortBenchmark.main: null with word counts: [10000]
2023-11-06 19:30:06 INFO  Benchmark_Timer - Begin run: intArraysorter with 100 runs
2023-11-06 19:30:07 INFO  TimeLogger - Raw time per run (mSec):  6.55
2023-11-06 19:30:07 INFO  TimeLogger - Normalized time per run (n log n):  .72
2023-11-06 19:30:07 INFO  Benchmark_Timer - Begin run: integerArraysorter with 100 runs
2023-11-06 19:30:10 INFO  TimeLogger - Raw time per run (mSec):  20.71
2023-11-06 19:30:10 INFO  TimeLogger - Normalized time per run (n log n):  2.28
2023-11-06 19:30:10 INFO  SortBenchmark - Beginning String sorts
2023-11-06 19:30:11 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-11-06 19:30:11 INFO  SortBenchmark - Testing pure sorts with 844 runs of sorting 10,000 words
2023-11-06 19:30:11 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using so
2023-11-06 19:30:11 INFO  Benchmark_Timer - Begin run: Helper for MergeSort: with 10000 elements with 844 runs
2023-11-06 19:30:13 INFO  SorterBenchmark - Without Instrumentation:
2023-11-06 19:30:13 INFO  TimeLogger - Raw time per run (mSec):  2.42
2023-11-06 19:30:13 INFO  TimeLogger - Normalized time per run (n log n):  3.41
2023-11-06 19:30:13 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using so
2023-11-06 19:30:13 INFO  Benchmark_Timer - Begin run: Helper for QuickSort dual pivot with 10000 elements with 844 runs
2023-11-06 19:30:15 INFO  SorterBenchmark - Without Instrumentation:
2023-11-06 19:30:15 INFO  TimeLogger - Raw time per run (mSec):  2.65
2023-11-06 19:30:15 INFO  TimeLogger - Normalized time per run (n log n):  3.73
2023-11-06 19:30:15 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using so
2023-11-06 19:30:15 INFO  Benchmark_Timer - Begin run: Helper for Heapsort with 10000 elements with 844 runs
2023-11-06 19:30:18 INFO  SorterBenchmark - Without Instrumentation:
2023-11-06 19:30:18 INFO  TimeLogger - Raw time per run (mSec):  3.07
2023-11-06 19:30:18 INFO  TimeLogger - Normalized time per run (n log n):  4.31
2023-11-06 19:30:18 INFO  SortBenchmark - Beginning LocalDateTime sorts

Process finished with exit code 0
```

Changing the config file to run all three sorts for the given number of words
Setting the array size in a custom config



Name: Sorting Config

□ Store as project file

java 19 SDK of 'INFO6205' mc

edu.neu.coe.info6205.util.SortBenchmark

10000

Working directory: /Users/abhishekguragol/Desktop/Fall23/INFO6205

Environment variables: