INFO 6205 - Program Structures and Algorithms

Spring 2024

NAME: Neha Devarapalli

NUID: 002883137

GITHUB LINK: https://github.com/nehadevarapalli/INFO6205

Task Assignment 6 (Hits as time predictor) :

To determine what the best predictor is of total execution time : comparisons, swaps/copies, hits (array accesses), or something else.

Relationship Conclusion:

In conclusion, the best predictor of total execution time for sorting algorithms would be the number of hits (the number of memory accesses / array accesses).

Evidence:

After looking at the below provided benchmark data consisting of mean hits, number of copies, mean swaps, mean compares, and their runtime in milliseconds with respect to the length of the input array, we can see that there seems to be a correlation between the memory accesses (hits) and the execution time, as seen in Heap Sort having the highest mean hits and henerally longer runtimes compared to the other algorithms. Also, I think that this correlation might not be linear. The number of comparisons and swaps are both very important factors influencing the runtime, but their impact may vary depending on the sorting algorithm. For instance, we can see that Quick Sort has more comparisons and swaps compared to Merge Sort but still its runtime is relatively lower. Now coming to the number of copies, I think copies contribute to the sorting algorithms becoming slower as we can see in the case of Merge Sort because there is obviously a memory access overhead caused by reading from one memory location and then writing to another which can cause latency.

Merge Sort (N)	Mean Hits	Copies	Mean Swaps	Mean Compares	Runtime (ms)
10000	269,783	110,000	9,761	121,503	2.84
20000	579,561	240,000	19,520	263,003	5.92
40000	1,239,131	520,000	39,043	566,008	12.81
80000	2,638,185	1,120,000	78,061	1,211,991	33.51
160000	5,596,409	2,400,000	156,130	2,583,994	67.85

Quick Sort Dual Pivot (N)	Mean Hits	Copies	Mean Swaps	Mean Compares	Runtime (ms)
10000	423,853	0	66,540	155,918	2.66
20000	915,593	0	142,489	340,550	5.59
40000	1,946,601	0	298,033	738,603	12.09
80000	4,212,998	0	654,778	1,572,033	25.99
160000	8,933,612	0	1,368,975	3,389,239	56.12

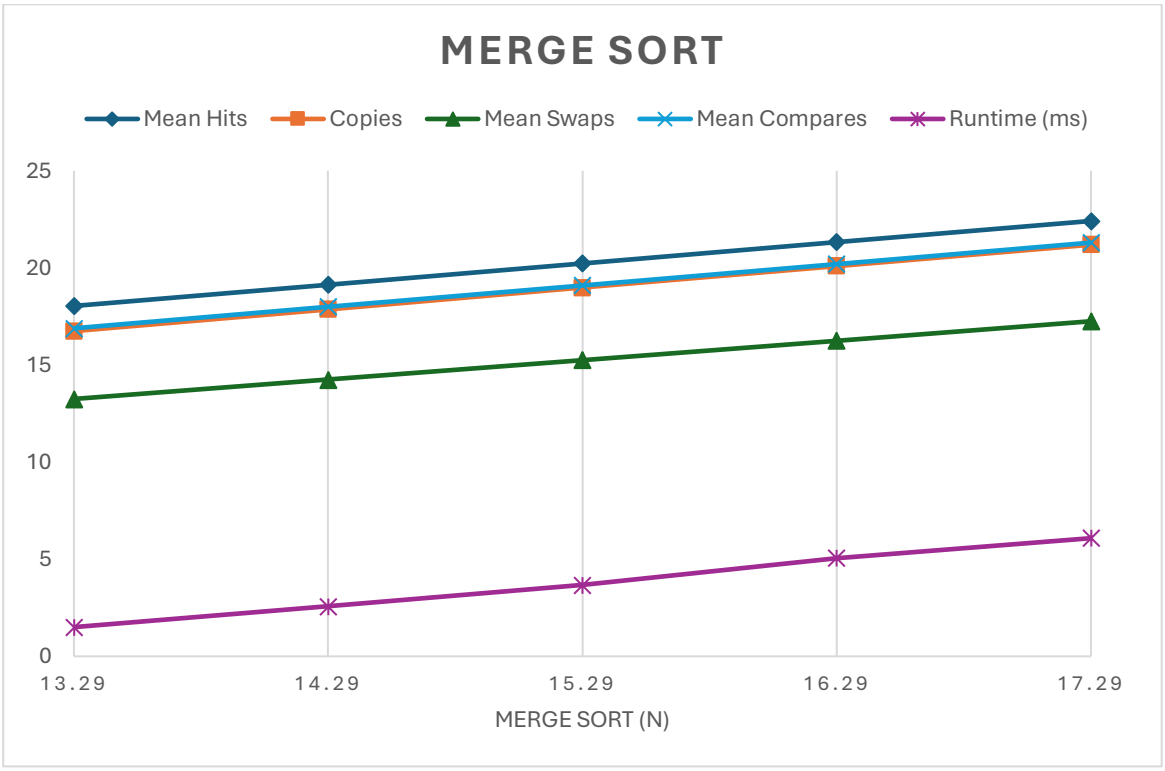
Heap Sort (N)	Mean Hits	Copies	Mean Swaps	Mean Compares	Runtime (ms)
10000	967,556	0	124,203	235,371	3.34
20000	2,095,056	0	268,396	510,736	7.37
40000	4,510,173	0	576,795	1,101,497	16.07
80000	9,660,290	0	1,233,593	2,362,959	37.46
160000	20,600,649	0	2,627,179	5,045,966	89.93

Even though, the hits are higher for QuickSort as compared to MergeSort, MergeSort is slower because of the high number of copies as we can see above.

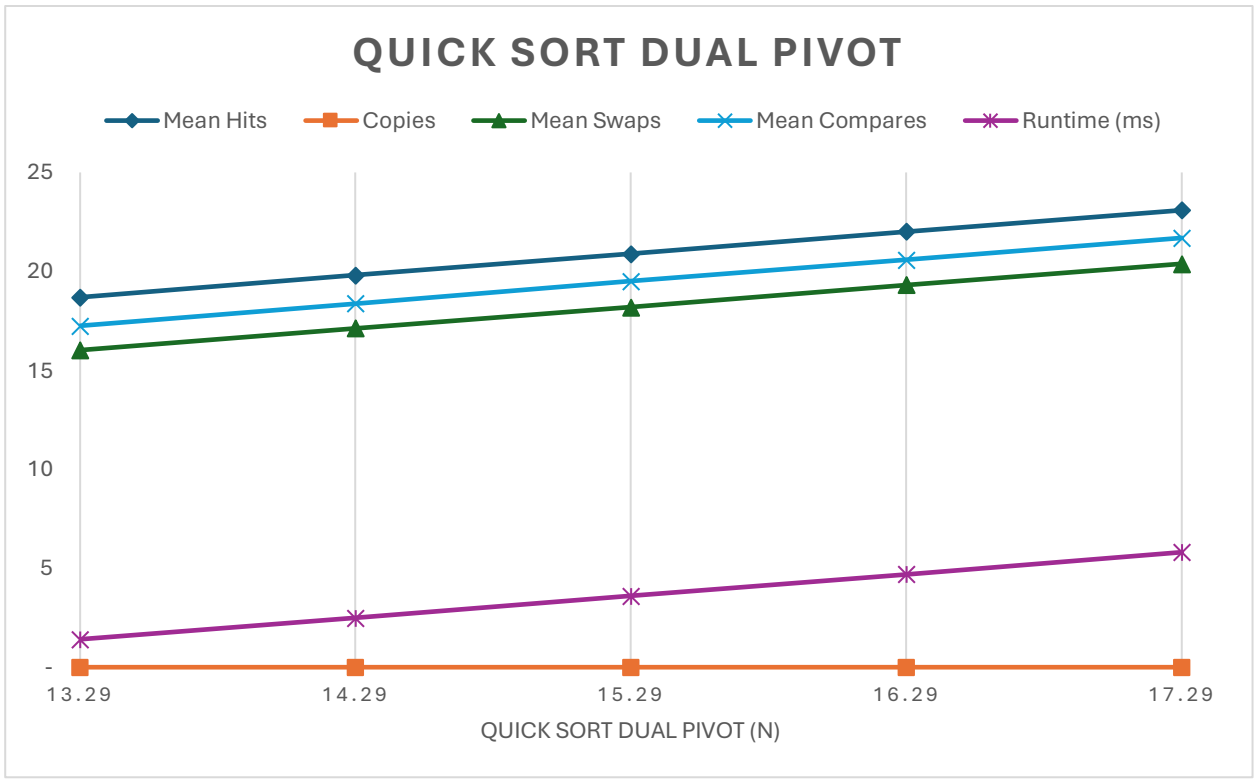
Graphical Representation:

The below given graphs are log/log charts of the above benchmark data. Here, both the axes are on the logarithmic scale.

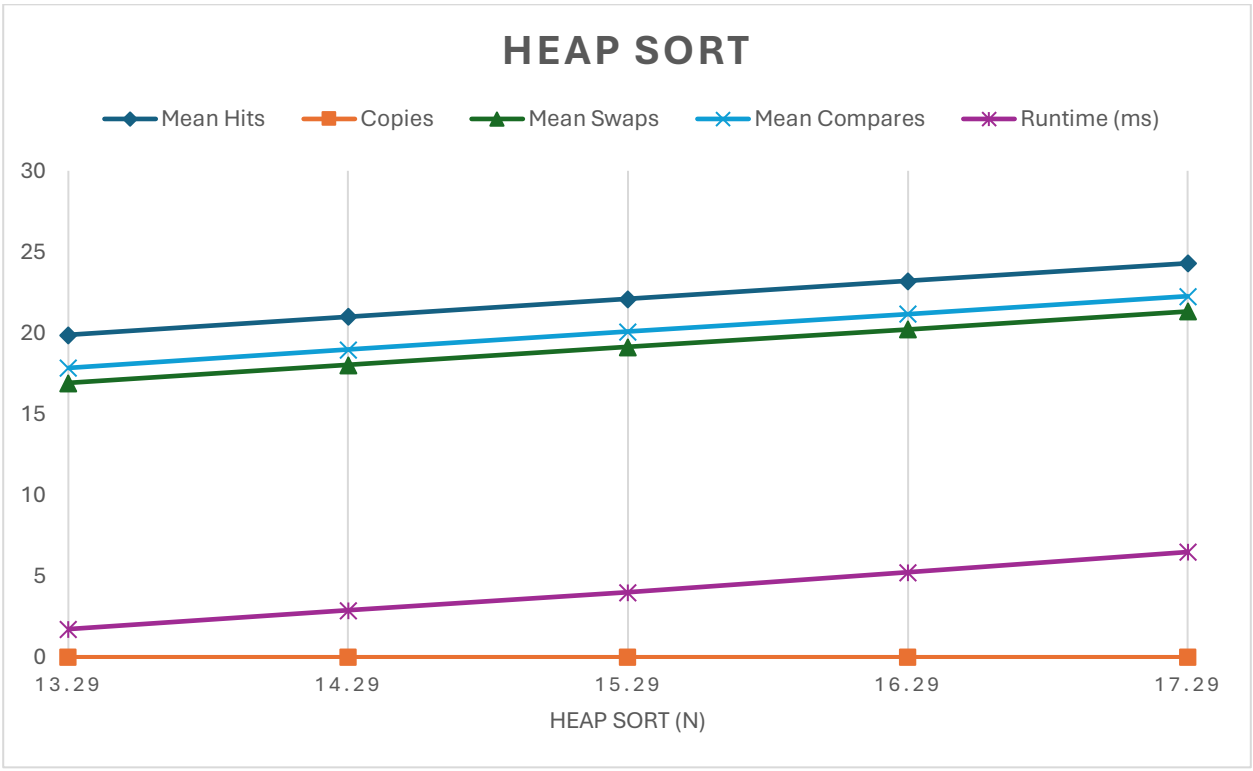
1. **Merge Sort**



2. **Quick Sort Dual Pivot**



3. **Heap Sort**



Output Screenshot: