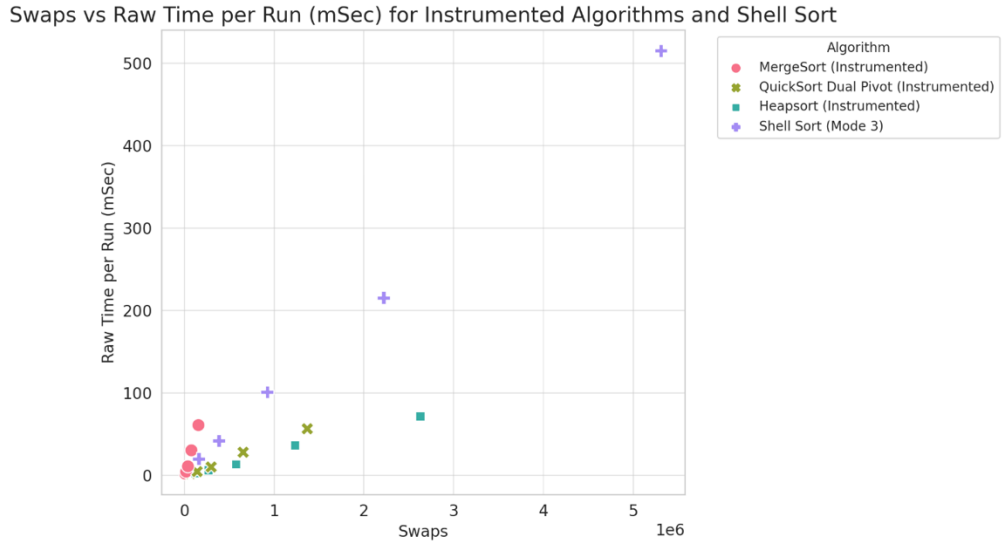
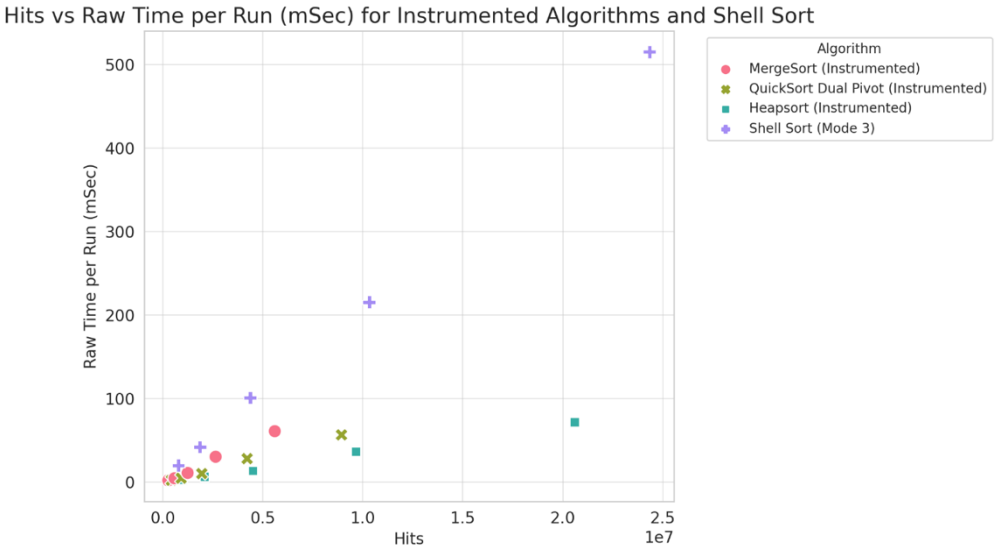
Program Structures and Algorithms
Spring 2024

NAME: Zenan Fan
NUID: 002854067
GITHUB LINK: https://github.com/LearningMach1ne/INFO6205

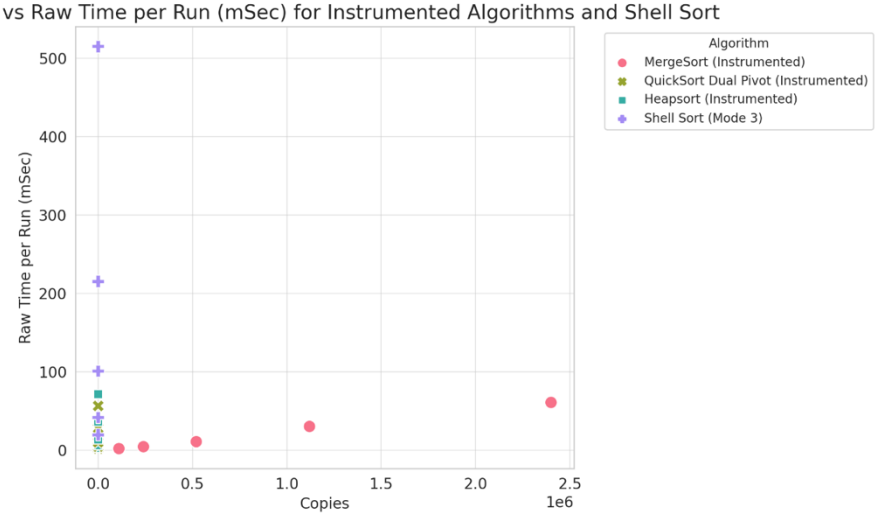
Task: Assignment 6 (Hits as time predictor)

Log chart:

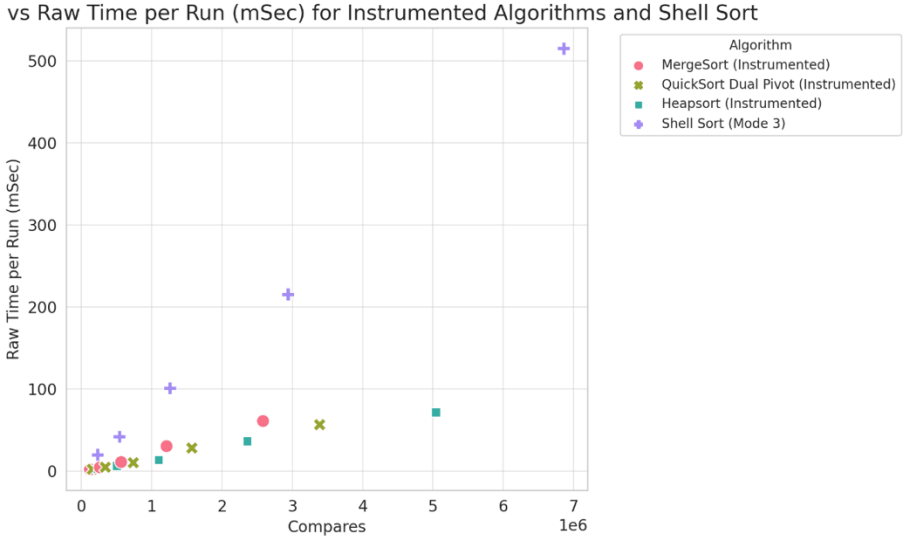




Copies vs Raw Time per Run (mSec) for Instrumented Algorithms and Shell Sort



Compares vs Raw Time per Run (mSec) for Instrumented Algorithms and Shell Sort



Spreadsheet:

	A	B	C	D	E	F	G	H	I
1	**Algorithm**	**Data Size**	**Raw Time per Run (mSec)**	**Normalized Time**	**Hits**	**Copies**	**Swaps**	**Compares**	
2	MergeSort	10,000	2.12	2.99	269,783	110,000	9,761	121,503	
3	QuickSort Dual Pivot	10,000	2.21	3.12	423,853	0	66,540	155,918	
4	Heapsort	10,000	2.74	3.86	967,556	0	124,203	235,371	
5	MergeSort (Instrumented)	10,000	1.99	2.81	269,783	110,000	9,761	121,503	
6	QuickSort Dual Pivot (Instrumented)	10,000	2.09	2.95	423,853	0	66,540	155,918	
7	Heapsort (Instrumented)	10,000	2.7	3.8	967,556	0	124,203	235,371	
8	MergeSort	20,000	4.65	3.02	579,561	240,000	19,520	263,003	
9	QuickSort Dual Pivot	20,000	4.51	2.93	915,593	0	142,489	340,550	
10	Heapsort	20,000	6.03	3.91	2,095,056	0	268,396	510,736	
11	MergeSort (Instrumented)	20,000	4.43	2.88	579,561	240,000	19,520	263,003	
12	QuickSort Dual Pivot (Instrumented)	20,000	4.54	2.95	915,593	0	142,489	340,550	
13	Heapsort (Instrumented)	20,000	5.98	3.88	2,095,056	0	268,396	510,736	
14	MergeSort	40,000	9.51	2.86	1,239,131	520,000	39,043	566,008	
15	QuickSort Dual Pivot	40,000	9.7	2.92	1,946,601	0	298,033	738,603	
16	Heapsort	40,000	13.05	3.93	4,510,173	0	576,795	1,101,497	
17	MergeSort (Instrumented)	40,000	10.87	3.27	1,239,131	520,000	39,043	566,008	
18	QuickSort Dual Pivot (Instrumented)	40,000	10.06	3.03	1,946,601	0	298,033	738,603	
19	Heapsort (Instrumented)	40,000	13.32	4.01	4,510,173	0	576,795	1,101,497	
20	MergeSort	80,000	23.71	3.33	2,638,185	1,120,000	78,061	1,211,991	
21	QuickSort Dual Pivot	80,000	22.4	3.14	4,212,998	0	654,778	1,572,033	
22	Heapsort	80,000	34.21	4.8	9,660,290	0	1,233,593	2,362,959	
23	MergeSort (Instrumented)	80,000	30.27	4.25	2,638,185	1,120,000	78,061	1,211,991	
24	QuickSort Dual Pivot (Instrumented)	80,000	27.97	3.93	4,212,998	0	654,778	1,572,033	
25	Heapsort (Instrumented)	80,000	36.38	5.1	9,660,290	0	1,233,593	2,362,959	
26	MergeSort	160,000	55.29	3.63	5,596,409	2,400,000	156,130	2,583,994	
27	QuickSort Dual Pivot	160,000	52.28	3.44	8,933,612	0	1,368,975	3,389,239	
28	Heapsort	160,000	77.96	5.13	20,600,649	0	2,627,179	5,045,966	
29	MergeSort (Instrumented)	160,000	60.9	4	5,596,409	2,400,000	156,130	2,583,994	
30	QuickSort Dual Pivot (Instrumented)	160,000	56.45	3.71	8,933,612	0	1,368,975	3,389,239	
31	Heapsort (Instrumented)	160,000	71.49	4.7	20,600,649	0	2,627,179	5,045,966	
32	Shell Sort (Mode 3)	10,000	19.58	90.88	791,349	0	162,388	233,287	
33	Shell Sort (Mode 3)	20,000	41.67	76.75	1,859,460	0	387,052	542,678	
34	Shell Sort (Mode 3)	40,000	100.79	73.68	4,376,107	0	924,370	1,263,684	
35	Shell Sort (Mode 3)	80,000	215.19	62.43	10,330,198	0	2,222,303	2,942,796	
36	Shell Sort (Mode 3)	160,000	514.99	59.29	24,350,272	0	5,311,248	6,863,888	
37									

Conclusion:

From the log chart we can see the runtime doesn't change much as the hit increases, and most algorithms in this experiment don`t use copies, the only algorithm using copies, merge sort, doesn`t affect much runtime when copies increase.

For the swaps, different algorithms have different result in the experiment, it has an important impact on the performance of merge sort, which is different from other algorithms. As can be seen from the figure of comparison, the running time varies with the number of comparisons, which indicates that the comparison operation has a significant impact on the performance of these algorithms. As a conclusion, the number of comparisons seems to be the best predictor of total execution time, as it plays a central role in almost all of algorithms.