

Assignment 6

1. Output

-----Test Starts -----

Merge Sort :

```
ArraySize = 10000
test: StatPack {hits: 560,636, normalized=6.087; copies: 80,000, normalized=0.869;
inversions: <unset>; swaps: 95,684, normalized=1.039; fixes: 25,250,513,
normalized=274.154; compares: 184,104, normalized=1.999}
Instrumenting helper for test with 10,000 elements
time: 20.561 msec
```

Quick sort Dual Pivot:

```
ArraySize = 10000
test: StatPack {hits: 520,280, normalized=5.649; copies: 0, normalized=0.000; inversions:
<unset>; swaps: 97,455, normalized=1.058; fixes: 26,189,665, normalized=284.351;
compares: 174,815, normalized=1.898}
Instrumenting helper for test with 10,000 elements
time: 166.4197999999998 msec
```

Heap Sort :

```
ArraySize = 10000
test: StatPack {hits: 967,204, normalized=10.501; copies: 0, normalized=0.000; inversions:
<unset>; swaps: 124,142, normalized=1.348; fixes: 75,320,790, normalized=817.785;
compares: 235,318, normalized=2.555}
Instrumenting helper for test with 10,000 elements
time: 293.8162 msec
```

Merge Sort :

```
ArraySize = 40000
test: StatPack {hits: 2,392,040, normalized=5.643; copies: 400,000, normalized=0.944;
inversions: <unset>; swaps: 380,191, normalized=0.897; fixes: 400,054,016,
normalized=943.823; compares: 813,773, normalized=1.920}
Instrumenting helper for test with 40,000 elements
time: 13.732299999999999 msec
```

Quick sort Dual Pivot:

```
ArraySize = 40000
test: StatPack {hits: 2,530,445, normalized=5.970; copies: 0, normalized=0.000; inversions:
<unset>; swaps: 451,354, normalized=1.065; fixes: 561,971,113, normalized=1325.824;
compares: 904,738, normalized=2.134}
Instrumenting helper for test with 40,000 elements
time: 6100.3555 msec
```

Heap Sort :

```
ArraySize = 40000
test: StatPack {hits: 4,510,400, normalized=10.641; copies: 0, normalized=0.000; inversions:
<unset>; swaps: 576,741, normalized=1.361; fixes: 1,213,112,303, normalized=2862.023;
compares: 1,101,718, normalized=2.599}
Instrumenting helper for test with 40,000 elements
time: 5137.1852 msec
```

Merge Sort :

```
ArraySize = 160000
test: StatPack {hits: 10,220,414, normalized=5.331; copies: 1,920,000, normalized=1.001;
inversions: <unset>; swaps: 1,523,744, normalized=0.795; fixes: 2,104,247,907,
normalized=1097.524; compares: 3,578,458, normalized=1.866}
Instrumenting helper for test with 160,000 elements
time: 43.8137 msec
```

Quick sort Dual Pivot:

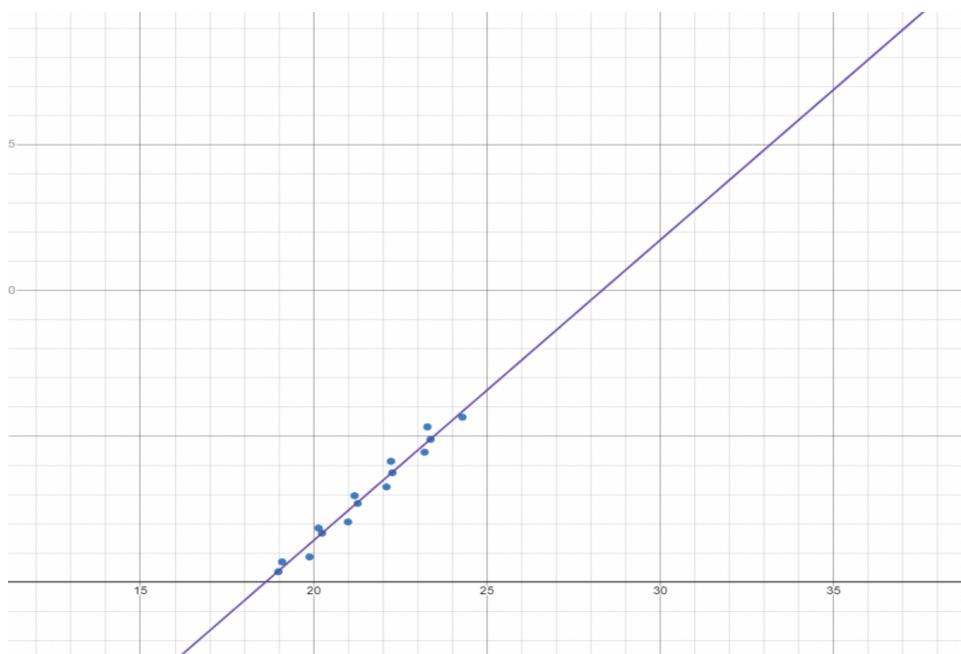
```
ArraySize = 160000
test: StatPack {hits: 10,914,490, normalized=5.693; copies: 0, normalized=0.000; inversions:
<unset>; swaps: 1,967,668, normalized=1.026; fixes: -1,805,863,067, normalized=-941.894;
compares: 3,765,183, normalized=1.964}
Instrumenting helper for test with 160,000 elements
time: 47362.7533 msec
```

Heap Sort :

```
ArraySize = 160000
test: StatPack {hits: 20,599,822, normalized=10.744; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 2,627,062, normalized=1.370; fixes: -2,090,718,764,
normalized=-1090.467; compares: 5,045,787, normalized=2.632}
Instrumenting helper for test with 160,000 elements
time: 99489.671 msec
```

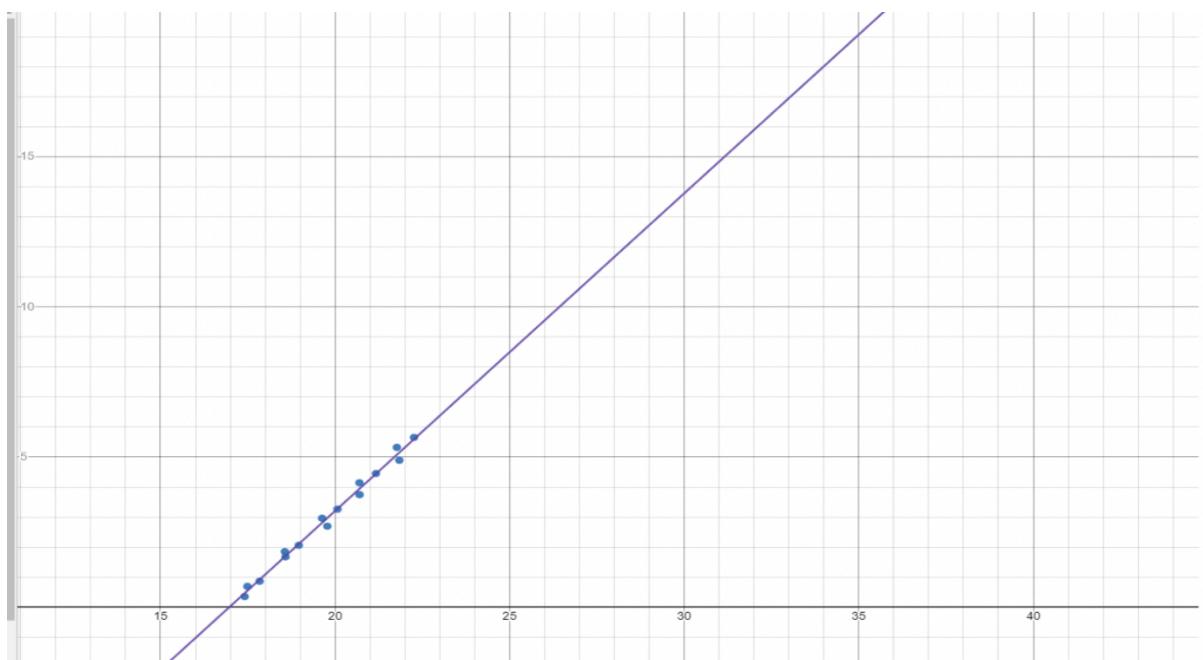
2. Observations

log(Hits)	log(Time)
19.09	0.69
18.98	0.35
19.88	0.86
20.14	1.85
20.24	1.85
20.24	1.67
20.99	2.06
21.18	2.96
21.27	2.69
22.1	3.26



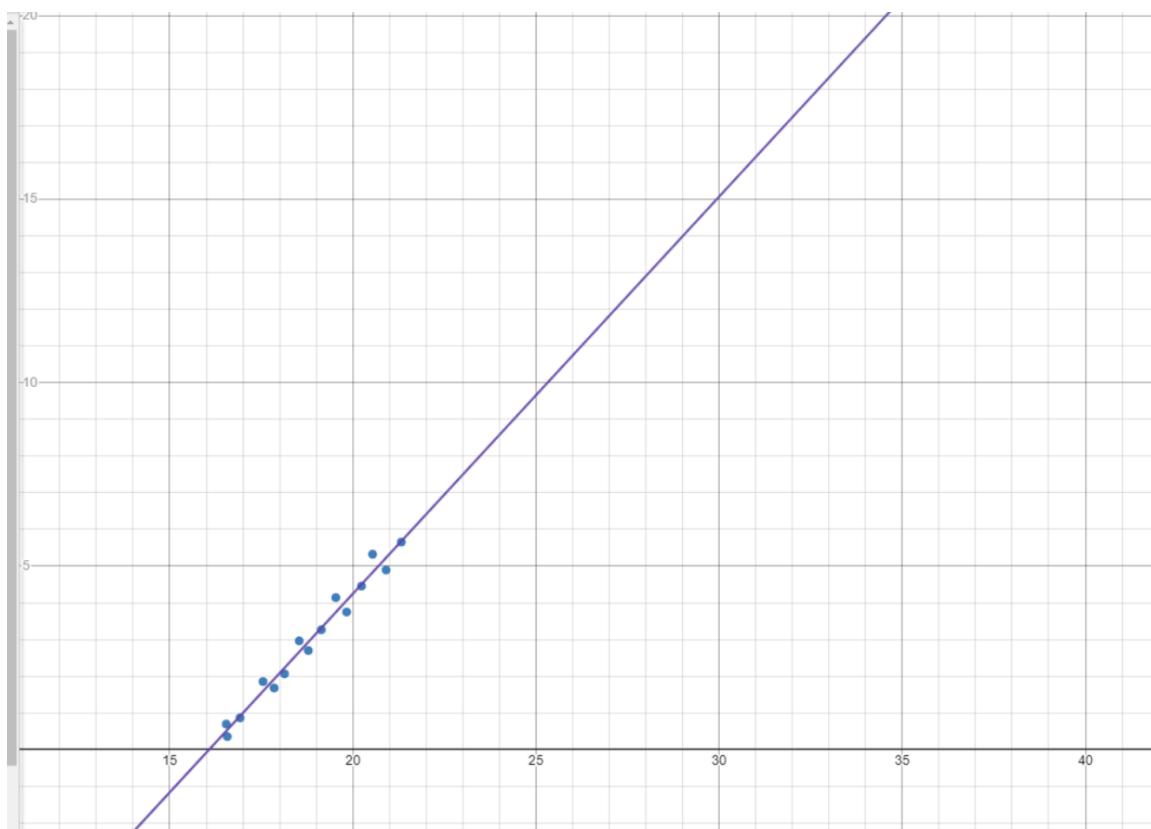
Log (hits) X log (time)

log(compare)	log(time)
17.49	0.69
17.41	0.35
17.84	0.86
18.56	1.85
18.58	1.67
18.96	2.06
19.63	2.96
19.78	2.69
20.07	3.26
20.7	4.1



Log (compare) X Log(Time)

$\log(\text{swap})$	$\log(\text{time})$
16.5	0.69
16.5	0.35
16.9	0.86
17.5	1.85
17.85	1.67
18.13	2.06
18.5	2.96
18.7	2.69
19.13	3.2
19.5	4.1



Log(swap) X Log(time)

3. Conclusion

Total execution time is directly proportional to the length of array. As per the observations, swap has the greatest slope which means swapping is the best predictor of total execution time than hits and comparisons.

Swaps > compares > hits