Program Structures and Algorithms Spring 2023(SEC- 3)

**Name: Altaf Salim Shaikh  
NUID: 002774748  
Assignment: 06**  
  
**Task**:  
Our task at hand is to run the benchmarks for merge sort, (dual-pivot) quick sort, and heap sort. You will sort randomly generated arrays of between 10,000 and 256,000 elements (doubling the size each time).  
  
**Code Snippets :**Text

Description automatically generated  
  
**Test Cases:**

Text

Description automatically generated

**Observations**: Here is the table for timings for three different types of sorts.

Table

Description automatically generated

Chart, line chart

Description automatically generated

This is a plot between the raw times of three sorts, this shows for this input merge sort shows significantly increasing trend.

Table

Description automatically generated

Chart, line chart

Description automatically generated

Table

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Table

Description automatically generatedChart, line chart

Description automatically generated

**Conclusion :**When evaluating the efficiency of sorting algorithms, the number of array hits resulting from operations such as compares, copies, and swaps can be used as a neutral way to determine which algorithm performs better. A larger number of hits implies poorer performance. However, if the operations differ in duration, the one that takes less time and involves fewer parameters is a better predictor of the algorithm's completion time. Swaps are more expensive than copies, which makes copy operations less costly.   
However, comparing copies and comparisons is not always straightforward, as the cost of comparisons can vary depending on the hardware. In terms of determining which algorithm performs worse, the one with the most swaps has the worst performance, followed by copy and comparison operations.

If no other metrics are available, a general number of hits can be used to determine the algorithm's performance, with the highest number indicating the worst performance. Merge sort has been observed to have the best performance, followed by quicksort and heapsort