**Program Structures and Algorithms**

**Spring 2023(SEC – 01)**

**NAME:** Neha Rastogi

**NUID:** 002709191

**ASSIGNMENT:** 6

**Task:** To determine for sorting algorithms namely Merge Sort, Dual Pivot Quick Sort, and Heap Sort what is the best predictor of total execution time: comparisons, swaps/copies, hits (array accesses), or something else

**Relationship Conclusion:**

The conclusion for relationship between comparisons, swaps/copies, hits (array accesses) as time predictors is as follows:

**HITS > COMPARISONS > SWAPS**

Based on the graphs and data obtained it can be concluded that, when compared to all other predictors that were taken into consideration, the predictor variable "Hits" has the biggest impact on the overall runtime of the program. This suggests that a key factor affecting how long a program takes to run is the number of hits, which are defined as the number of times a specific element or value is accessed during program execution. Every time we require an access to the memory it takes up time and retrieving the values adds to the overall time taken up.

Also, although not as strongly as the "hits" variable, it was found that the predictor "comparisons" also significantly affects the program execution time. This implies that the entire execution time of a program is also influenced by the quantity of comparisons between values. To optimize the program, reducing the times memory must be accessed will directly reduce total time taken by the sort.

Since Heap Sort has the maximum number of hits it has highest runtimes and MergeSort and DualPivotQuickSort have similar values for hits leading to similar time taken for sorting.

**Evidence to support that conclusion:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Graphical Representation:** Plotting a graph of the 3 sorting algorithms’ run time against Array size we see that the time taken by Merge Sort and Dual Pivot Quick Sort is comparable while that taken Heap Sort is higher than both.

For MergeSortBasic , below is the data followed by graph for time against array size followed by comparisons, swaps, and hits against array size



For DualPivotQuickSort, below is the data followed by graph for time against array size followed by comparisons, swaps, and hits against array size



For HeapSort, below is the data followed by graph for time against array size followed by comparisons, swaps, and hits against array size



Coming to comparisons, swaps, and hits (array accesses) we see that hits, that is the number of array accesses takes the maximum amount of time and impacts the total runtime the most. Therefore, there is a direct relation between the two values as is seen for an increasing size array that is generated randomly. Further comparisons also have a significant impact because it also involves array accesses although they are lesser in number thus making the impact lesser.

**Unit Test Screenshots:**

**A screenshot of a computer

Description automatically generatedText

Description automatically generated**