Project Analysis – QuickSort vs. MergeSort

The following are the specific observations from the results obtained by running QuickSort, MergeSort, and both programs concurrently. To produce this analysis, and to better evaluate the program and its benchmarks, the program was run with input sizes of 1,000,000, 2,000,000, and 3,000,000 using lists of random numbers.

* Runtime: as input size increased, so did the runtime for all three conditions. However, for very low input sizes, the program running individually performed better than concurrently. For greater input sizes, concurrently running the programs performed a lot better than individually. For most of the input sizes, MergeSort performed better, but for very large input sizes, QuickSort outperformed it.
* CPU Usage: with increasing input size, the CPU usage did not increase. Rather, it remained constant with some slight fluctuations. For very large input sizes, MergeSort was the one that had more CPU usage.
* Memory Usage: with increasing input size, memory usage increased a little bit. However, running both programs concurrently utilized a lot more memory than running them individually. QuickSort utilized a little bit more memory than MergeSort.
* Hard Drive Usage: this metric was constant throughout every input size for all running conditions.
* RSS: with increasing input size, RSS also increased for all conditions. MergeSort always had more RSS than QuickSort, but as input continually increased, running both programs concurrently soon had more RSS than when MergeSort ran alone.
* VMS: this metric was constant throughout every input size for all running conditions.
* Page Faults: with increasing input size, so did the number of page faults increase for all conditions. Overall, running both programs concurrently always had more page faults, and running MergeSort alone had more than running QuickSort alone.

The updated source code and this document can also be obtained from the GitHub repository at <https://github.com/cristianocaon/AOS_project.caon>

**Results:**

Input size: 1,000,000

Table

Description automatically generated

Input size: 2,000,000

Table

Description automatically generated with medium confidence

Input size: 3,000,000

Table

Description automatically generated