**Merge Sort**

|  |  |  |  |
| --- | --- | --- | --- |
| **Run No** | **1(1000 integers)** | **2(1500 integers)** | **3(2000 integers)** |
| **Merge Sort** | 2.9960s  2.9790s  4.2490s | 2.7360s  3.6290s  3.8310s | 3.0450s  3.4890s  2.0110s |
| **Average Tome Taken** | 3.408s | 3.398s | 2.8483s |

**Quick Sort**

|  |  |  |  |
| --- | --- | --- | --- |
| **Run No** | **1(1000 integers)** | **2(1500 integers)** | **3(2000 integers)** |
| **Quick Sort** | 3.519s  3.263s  2.538s | 3.569s  3.293s  2.580s | 3.283s  2.477s  3.943s |
| **Average Tome Taken** | 3.106s | 3.147s | 3.234s |

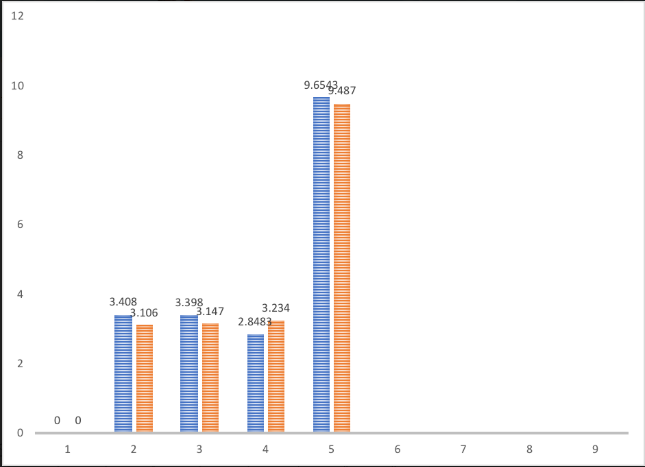
**Q)** Compares with theoretical time complexity? Explain any differences between the experimental and theoretical results

**Merge Sort**

* The theoretical time complexity of Merge Sort for Best, Worst and Average case is O(n log n).
* The experimental time complexity for Merge Sort for best case of 1000 integers is 3.408s.
* The experimental time complexity for Merge Sort for best case of 1500 integers is 3.498s.
* The experimental time complexity for Merge Sort for best case of 2000 integers is 2.848s.
* The experimental time complexity for Merge Sort for Worst case of 1000 integers is 3000ms(3s)
* The experimental time complexity for Merge Sort for Worst case of 1500 integers is 4764.13ms(4.764s).
* The experimental time complexity for Merge Sort for Worst case of 2000 integers is 6602.05ms(6.02s).

**Quick Sort**

* The theoretical time complexity of Merge Sort for Best and Average case is O(n log n).
* The theoretical time complexity of Merge Sort for Worst case is O(n ^2).
* The experimental time complexity for Quick Sort for best case of 1000 integers is 3.106s.
* The experimental time complexity for Quick Sort for best case of 1500 integers is 3.147s.
* The experimental time complexity for Quick Sort for best case of 2000 integers is 3.234s.
* The experimental time complexity for Quick Sort for Worst case of 1000 integers is 3000(3s)
* The experimental time complexity for Quick Sort for Worst case of 1500 integers is 4764.13ms(4.764s).
* The experimental time complexity for Quick Sort for Worst case of 2000 integers is 4000000ms(4000s)



**Q)**  Compare and contrast the results between merge sort and quick sort.  Explain anomalies if any.

**Quick Sort**

* The worst case of Quick Sort is O(n^2) compared to that of Merge Sort which has O(n log n)
* Faster than Merge sort for a small set of data.
* It is inefficient for larger number of elements.
* Quick Sort requires less storage memory compared to Merge Sort

**Merge Sort**

* The worst case of Merge Sort is O(n log n) compared to that of Quick Sort which has O(n^2).
* Same speed for all types of data.
* It is more efficient than Quick Sort.
* Merge Sort requires more storage memory.

**Anomalies**

* Quicksort has O(*n*2) worst-case runtime compared to the worst-case runtime of Mergesort which has O(*n*log*n*). However, it’s superior to merge sort in many scenarios because many factors influence an algorithm’s runtime, and, when taking them all together, quicksort wins out
* To avoid Quick Sort’s worst-case run time, we can use an appropriate choice of the pivot, such as picking it at random location or by picking the last element as pivot, so by this way Quick Sort runs faster than Merge Sort.