$\begin{array}{c} CS2810 \\ Advanced \ Programming \ Lab \\ Lab-2 \end{array}$

REPORT
FOR
QUESTION - 1

J S SUHAS CS13B056

System Configuration

Processor: AMD A10-4600M APU with Radeon HD7660G

RAM : 8.0 GB (7.47 available)

Hard Disk: 1 TB – 7200 rpm drive

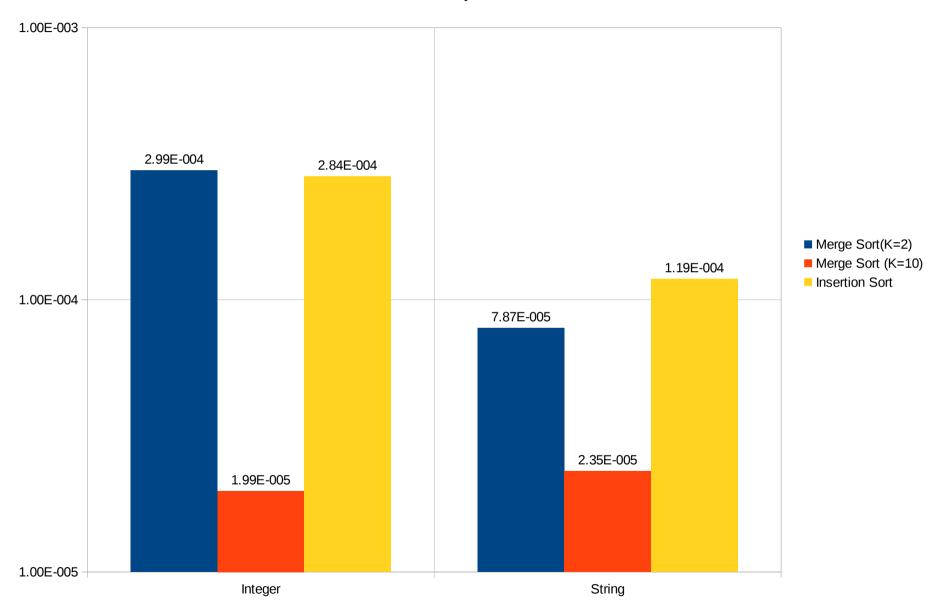
OS: Kubuntu 14.10 with KDE Plasma 4

Eclipse : Eclipse Luna (4.4.1)

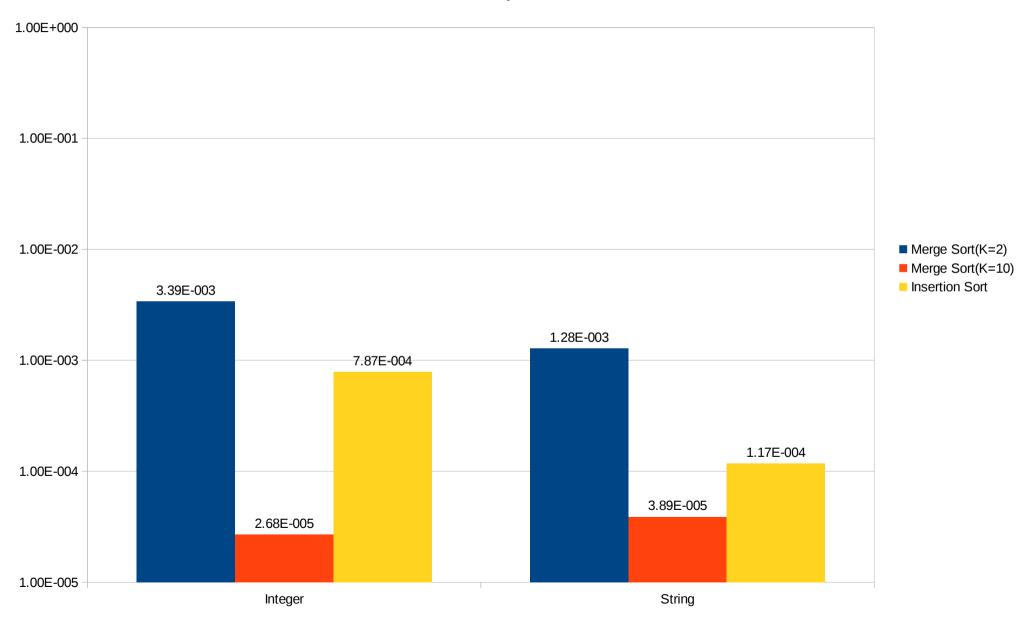
Java : Open JDK 7

Note: All y-axis values are in 'seconds' for all the graphs.

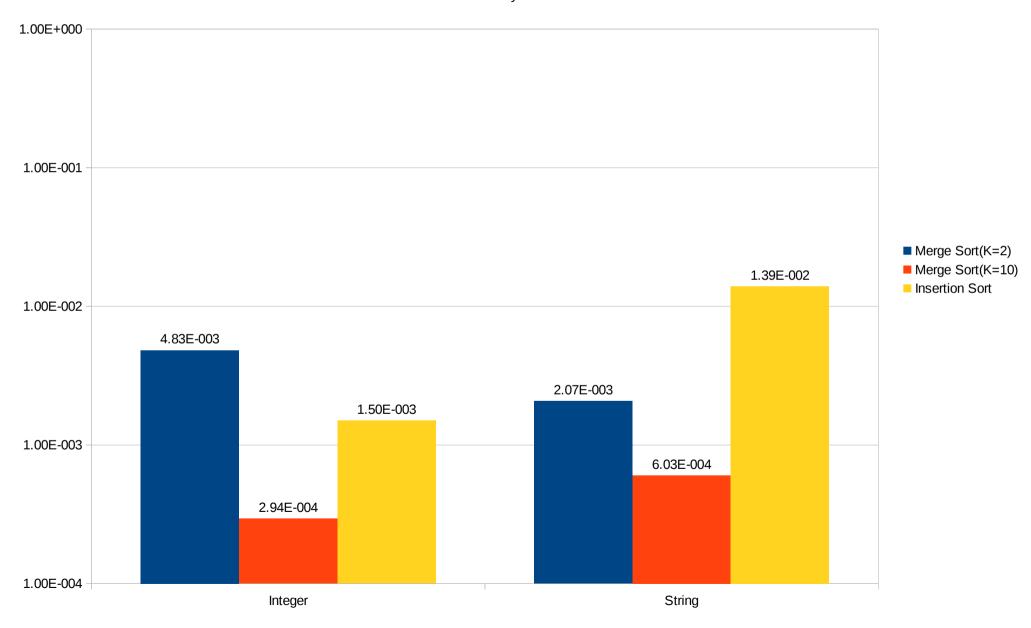
Array List



Array List

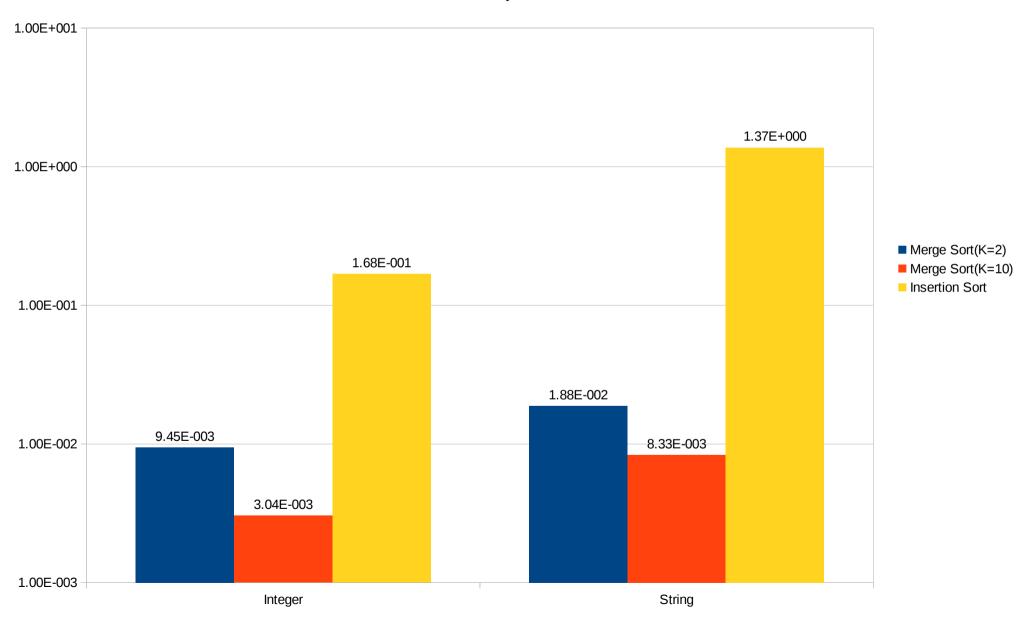


Array List



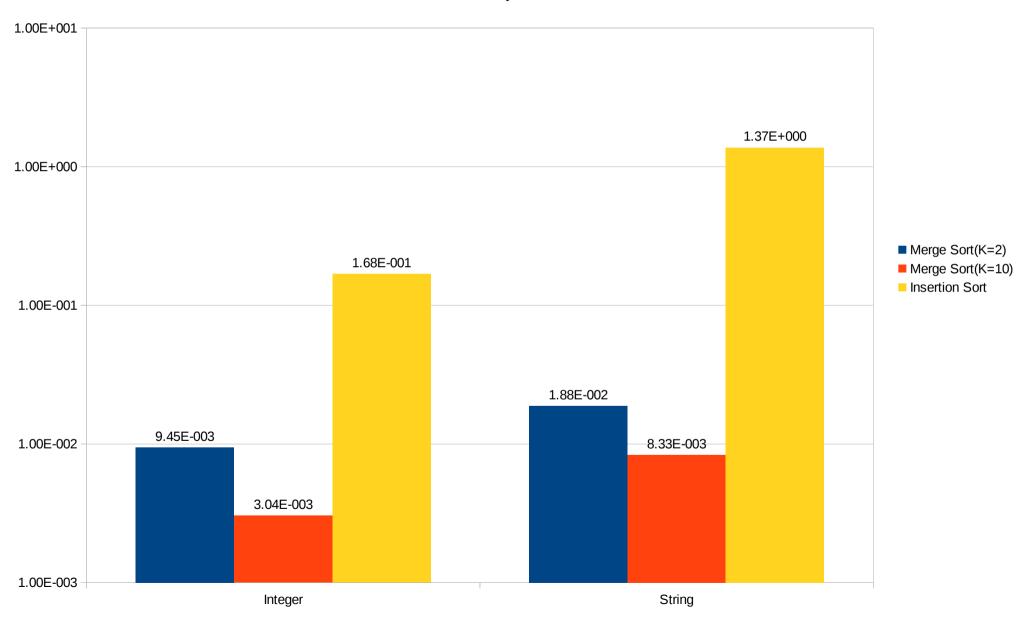
Sorting - Slze = 10,000





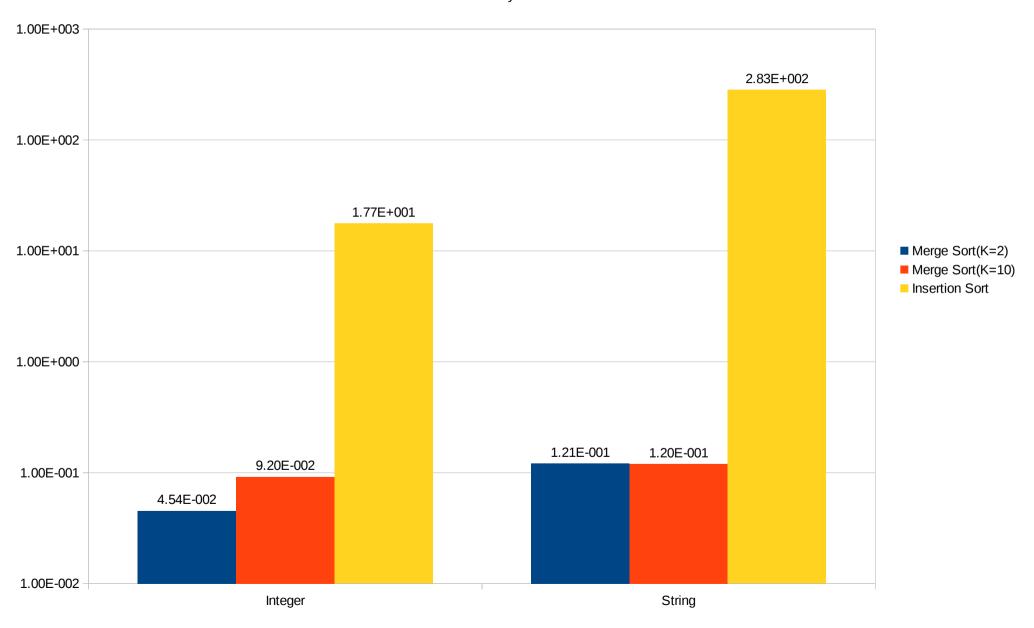
Sorting - Slze = 10,000





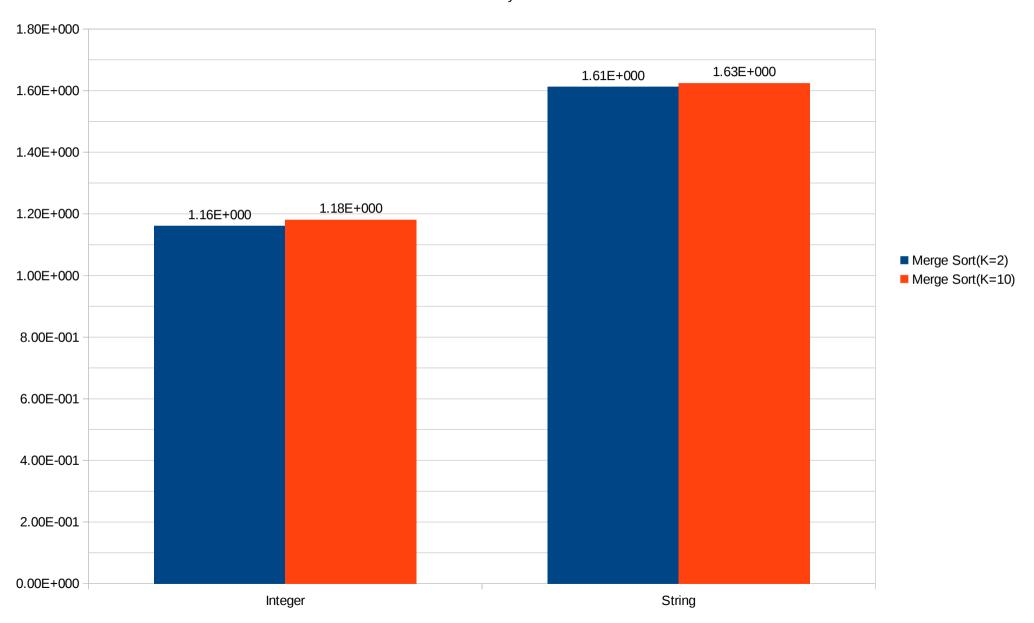
Sorting - SIze = 100,000

Array List

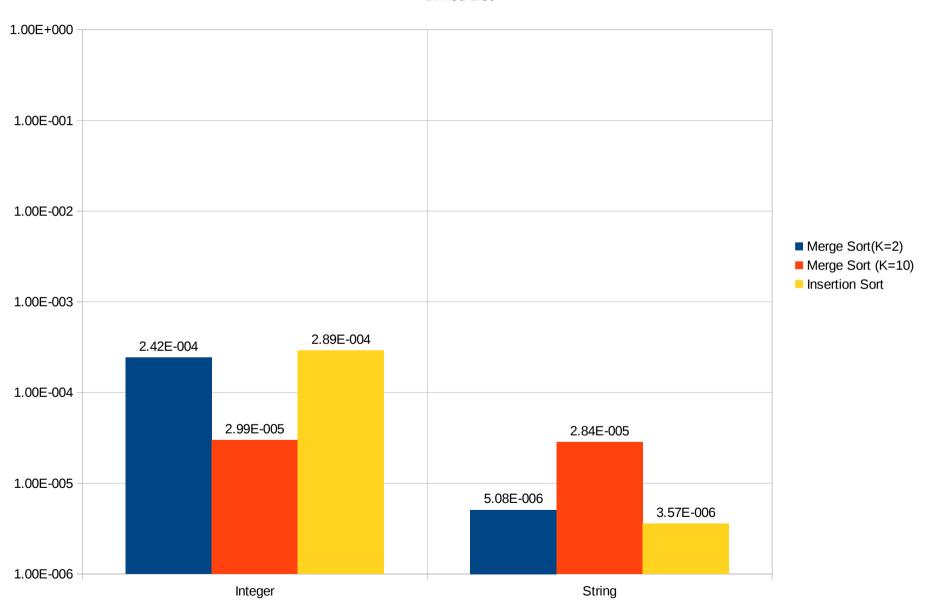


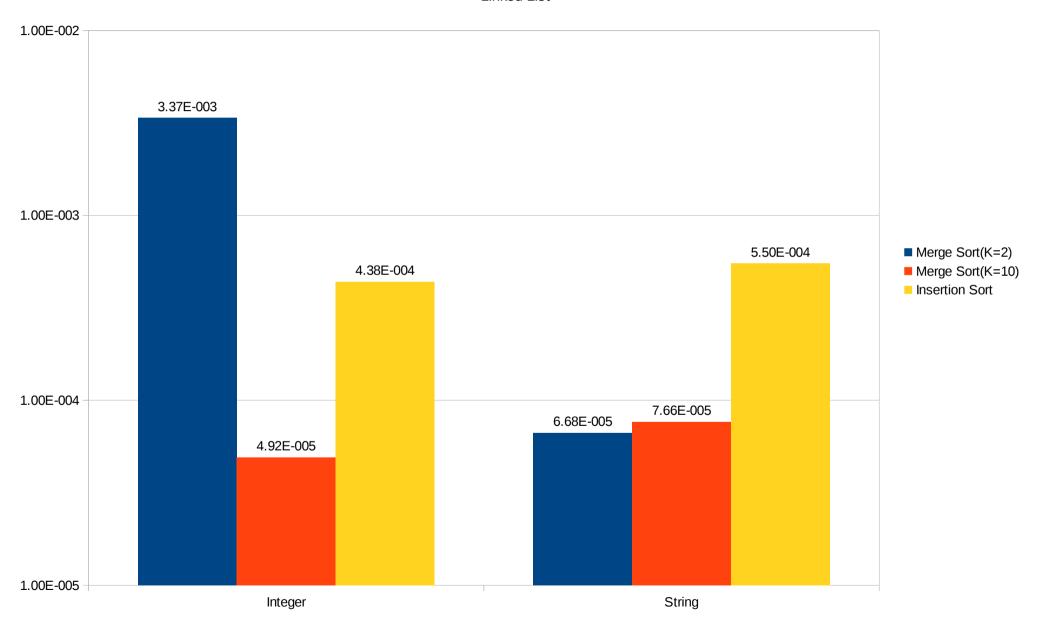
Sorting - SIze = 1,000,000

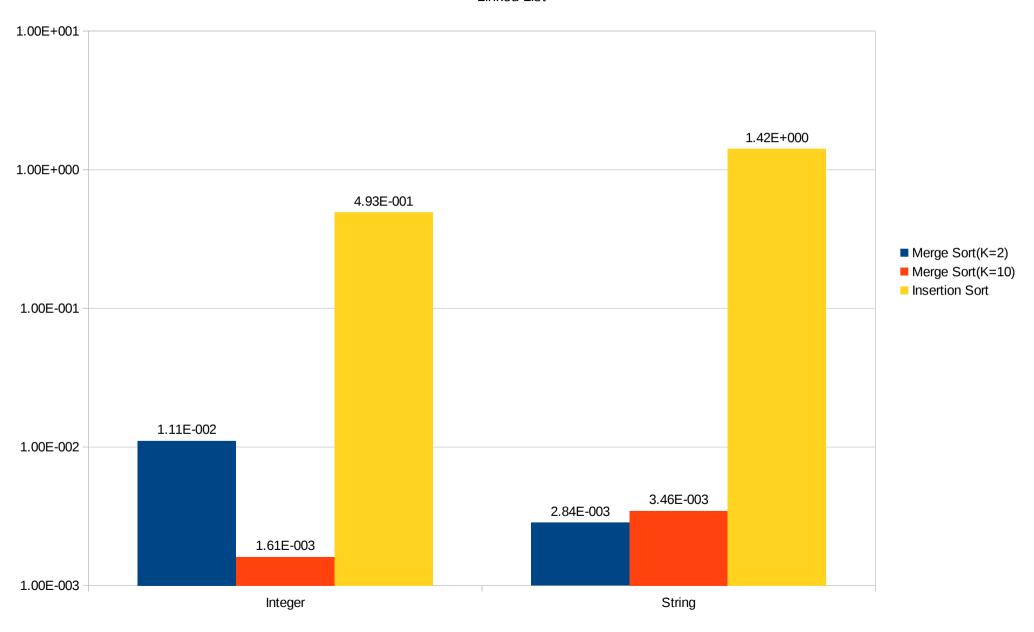
Array List



*Insertion sort ommitted for testing 1,000,000 size input

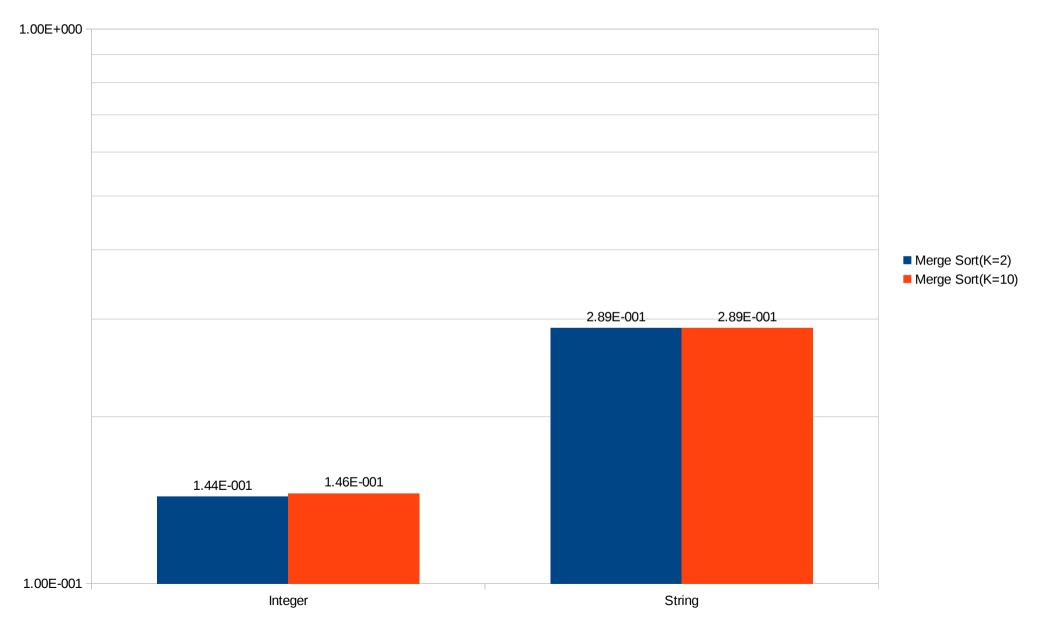






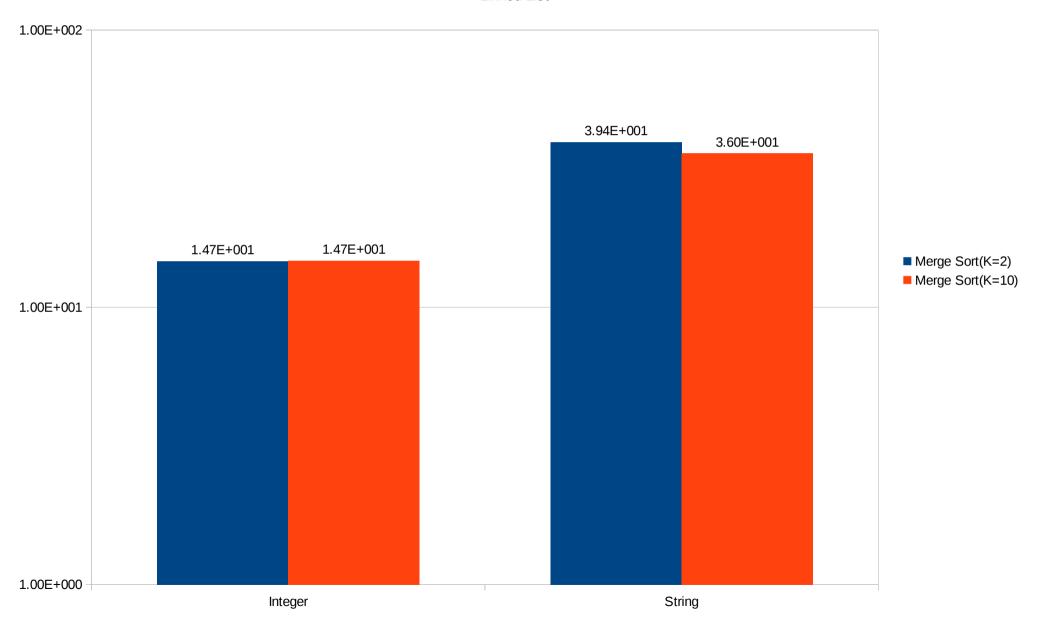
Sorting - SIze = 10,000





*Insertion sort ommitted for testing 10,000 size input

Sorting - SIze = 100,000



*Insertion sort ommitted for testing 100,000 size input

Observations

In general, the following observations could be drawn from the previous graphs:

- 1. Insertion sort fares very badly compared to Merge Sort and Hybrid Merge Sort (Merge Sort Bottoming out at K = 10). The performance is so poor that it was not feasible to run even a single Insertion Sort on 1,000,000 numbers when using Array List implementation.
- 2. Between Merge Sort and Hybrid Merge Sort, the latter takes a very small lead as the input size increases. The difference is not very significant.
- 3. The performance order of the algorithms is almost as theoretically predicted. The slight deviations could be due to implementation overheads.
- 4. In general, the String array sorting takes a lot more time than that of Integer Array sorting, using any sorting algorithm. The difference in performance is huge as is seen in the graphs above.
- 5. In general, Array List implementation of the sorting algorithms proved to be much faster than Linked List counterpart. This could be attributed to the additional overhead involved in accessing an element in Linked List (O(n) in Linked List, O(1) in Array List). This leads to poor performance while using Linked List implementation for a List.
- 6. On an average, $O(n^2)$ sorting algorithms, like Insertion Sort, runs like $O(n^3)$ when using Linked List implementation and O(n.log(n)) algorithms (Merge Sort) run like $O(n^2.log(n))$.
- 7. Overall, it could be said that Linked List implementation proves to be slower than Array List implementation and Integer sorting incurs smaller overheads than String sorting.