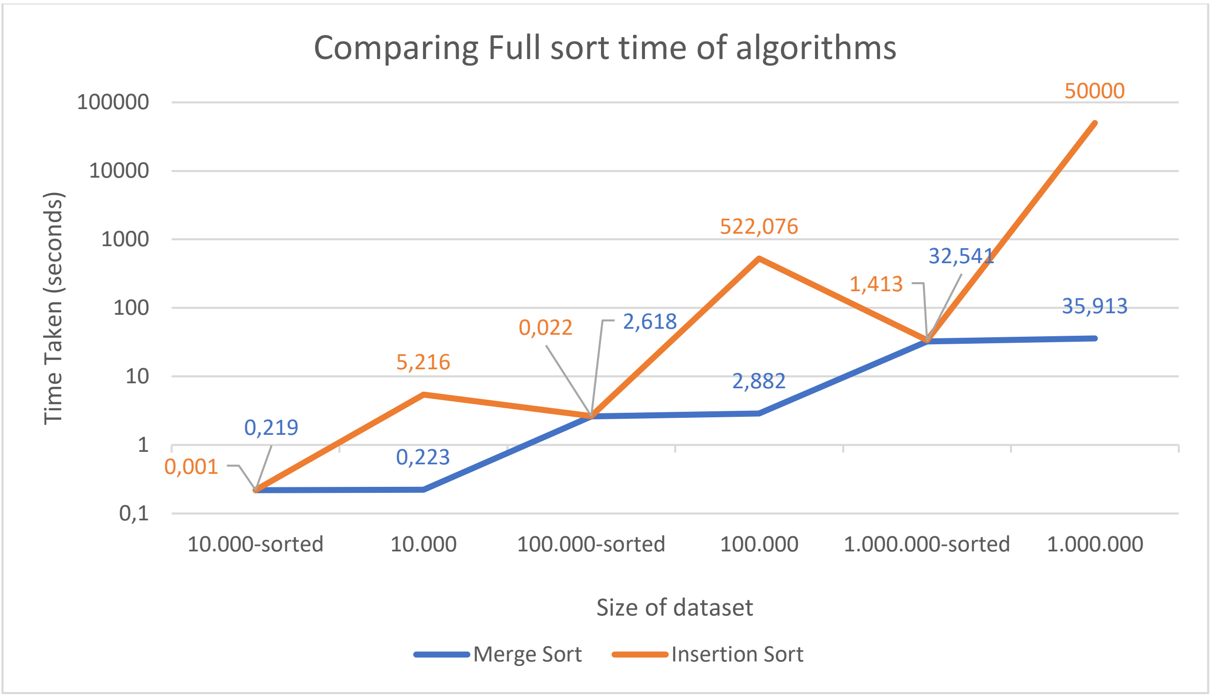
BLG35E – Analysis of Algorithms 1, Fall 2017

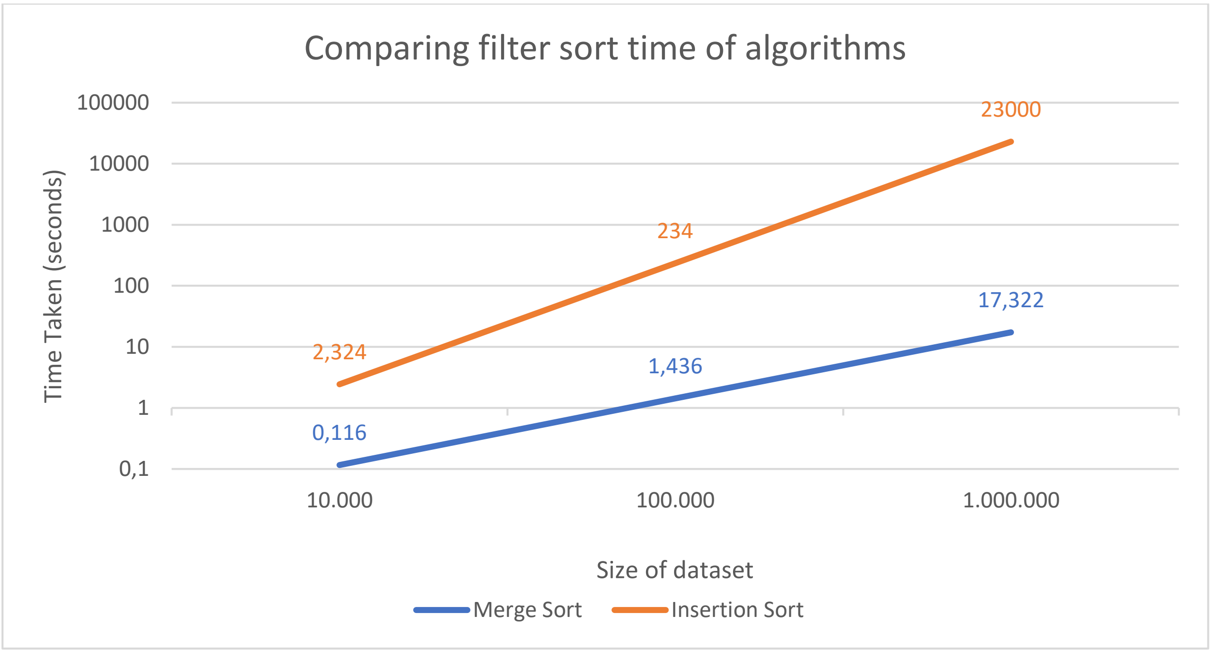
Project 1

Yunus Güngör 150150701

Compiling: compiled with command: `g++ main.cpp -std=c++11`

Q1. Comparing sorting algorithm's runtime





Graphic's time values are logarithmic. Values of insertion sort with dataset size of 1.000.000 are approximations. After an hour work time insertion sort process on dataset with 1.000.000 data was terminated. Given values are approximations according to algorithm's complexity and time taken on datasets with 10.000 data and 100.000 data.

Q2. Analysis of computational performances

Since insertion sort has complexity of $O(n^2)$ and merge sort has complexity $O(nlog(n))$, merge sort expected to run faster than insertion sort in most of the cases. But since insertion sort has complexity of $\theta(n)$ in best case, sorting already sorted datasets will take less time with insertion sort. Experiments done on datasets proves this point since merge sort takes less time to run in every unsorted dataset and insertion sort takes less time to run on sorted datasets.

Q3. **Analysis of dataset**

Dataset has different distributions for name, class, rarity, set, type and cost. Name has the most scattered values in dataset. Class, rarity, set, type and cost are rather fixed with at most 10 different values. Considering the case that sorting done by rarity or by set rather than type in filter sort, performance would not be affected much since all the values except name has a close distribution to each other. Since name has scattered values, sorting will be low performance especially in insertion sort because, more switch operation will be required.

Q4. **Stability**

Stable sorting means that, after sorting two elements with equal value, those elements keep their order at each output of the sort algorithm. That means, if there are two same values, values will keep their positions no matter how many times they are sorted. On the other hand, unstable algorithms might give different order for same values each time sorted. Some algorithms like insertion sort and merge sort are naturally stable. This stability also means that sorting algorithms can be stacked. That means first sorting by a value than sorting by another value in the previously sorted list is possible without any further do.