Analysis Of Algorithms
Homework 1 Report

150130032 – Baran Kaya

Q1)

Nanoseconds	Full/Insertion	Filter/Insertion	Full/Merge	Filter/Merge
10k	170575518961 ns	66707740316 ns	4010782998 ns	4027709821 ns
10k-sorted	264789622 ns	55947539576 ns	3955882697 ns	4008143185 ns
100k	More than 10mins	More than 10mins	49118775144 ns	49202568049 ns
100k-sorted	3606916720 ns	More than 10mins	48839436172 ns	49043213691 ns
1m	More than 10mins	More than 10mins	More than 10mins	More than 10mins
1m-sorted	140428563511 ns	More than 10mins	581677785217 ns	More than 10mins

*More than 10 mins = Reading + sorting takes more than 10 minutes and i stoped the program after 10 minutes.

Q2) When the input is sorted full insertion sort is better than the merge sort however, on filter catagory merge sort is way better than the insertion sort. Full insertion sort is better than merge on sorted input because insertion sort just checks all inputs and does nothing but merge sort divides and merges the input so it takes more time. Overall, merge sort is better than insertion because insertion has loop in a loop and it takes a lot of time.

Q3) Number of card names is the highest among the other card parameters like rarirt and set. So if we use name instead of rarity or set it will take more time and decrease the programs performance significantly.

Q4) Insertion sort that i used in this homework is stable because when 2 elements are equal this sort does not change their position, it changes their position only one is bigger than the other (<). However, in merge sort we cannot tell it because merge sort divides the array. But if the equal elements are in last part of dividing (just one element arrays) than it does not change their position either.

Extra

- I did not use chrono library for measuring time as nanosecond because SSH compiler did not support it. So i decided to use time() function therefore screen time may not be good for comparision. I measured some of them (max 10 mins.) and wrote them to the Q1.