MOHAMMAD IMMAM

SORTING ANALYSIS

COP3530

Dr. Cheryl Resch

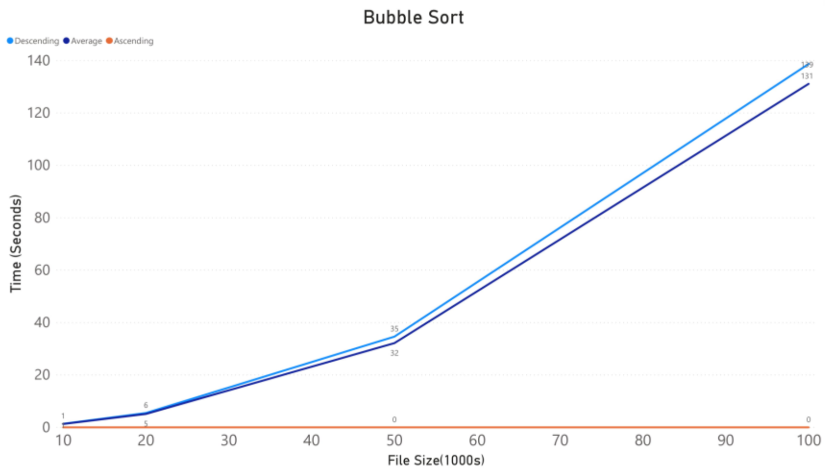
Introduction

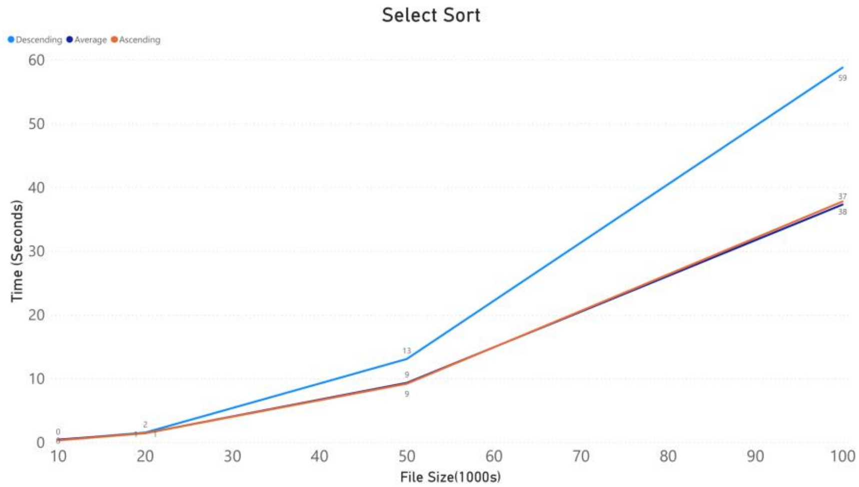
This document provides an analysis on Bubble Sort, Selection Sort and Merge Sort.

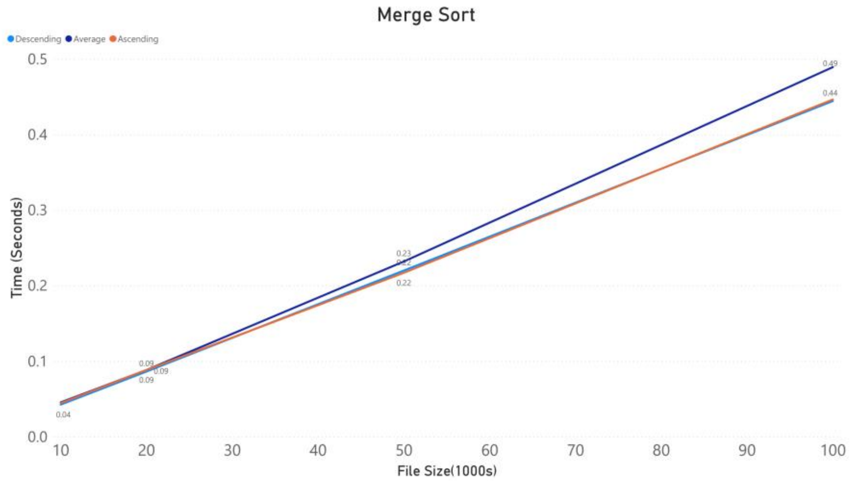
Method

C++ and its chrono library were used to time the sorts and compare. Each sort was tested 3 times to take an average for the records below. The sorting algorithms and the helper methods used are attached to the assignment. All the sorting was done on vector of integers. Excel was used to generate 10K, 20K, 50K and 100K integer files. The timing data was then, input in data queries of PowerBI to generate the scatterplots of Time in seconds vs File Size in 1000s.

Scatter Plot







Analysis

The selection sort has an average run time complexity of O(n^2). It has a slightly greater time requirement for worst case and takes almost the same time for best and average case. As the file size grow, the time required to sort these files grow exponentially. 100,000 files at worst case gets sorted in around 60 seconds. However, it takes around 40 seconds to sort a list even at best case which might not be favorable.

Bubble sort also grows quadratically as file size grows. At worst case it has computational complexity of O(n^2). However, the actual time requirement for bubble sort is quite high compared to the others except at best case in which it is O(n), just passing through the list to make sure it is sorted. Even with 100,000 files the time required to sort the list is negligible. Therefore, it is a favorable sort if we know the list will be nearly sorted.

Merge sort is comparatively the quickest sort. With computational complexity of O(n). At all the cases regarding the files, merge sort does not require more than a second at sorting even 100,000 integers. Looking at it comparatively, merge sort has a slight exception. It takes almost the same time to sort best and worst case but takes the most time to sort an average case.

Overall

Difference in performance: Merge Sort performs better comparatively regardless of size and cases. It is a favorable and go to sorting algorithm for a random list. Bubble sort performs good only on cases where list is almost sorted with few abnormal.

Linear/Quadratic: All the plots for selection sort looks quadratic since most of the times, for each item the list needs to be run though from beginning to end. Bubble sort plots look quadratic for the same reason except the base case which is linear since just one run is enough to make sure the list is sorted.

Affect of order: Ascending order means the list is already sorted and is the best case for any sorting algorithm. Descending lists are completely opposite, where each item requires a swap and many comparisons and proves to be the worst case. Random order is as the name suggests randomly distributed list and can account for an average case for an algorithm.