Assignment 1 Analysis

SFWR ENG 2XB3

Syed Aamir Tahir - 400134432

# Table of Sorting Algorithm Execution Times

The table of execution time data was broken into 4 separate tables, each depicting one type of sorting algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Size of Dataset | sortInsert Execution Time | sortComparable Execution Time | sortBinary Execution Time |
| 16 | 800 | 28800 | 3100 |
| 64 | 2700 | 31700 | 12500 |
| 256 | 77900 | 97600 | 86700 |
| 1024 | 2116800 | 859200 | 342400 |
| 4096 | 11705100 | 22875400 | 1983000 |

Table Insertion Sort Execution Times

|  |  |  |
| --- | --- | --- |
| Size of Dataset | sortMergeTD Execution Time | sortMergeBU Execution Time |
| 16 | 6400 | 3400 |
| 64 | 22300 | 10600 |
| 256 | 117600 | 46700 |
| 1024 | 551100 | 206900 |
| 4096 | 3760000 | 1186700 |

Table Merge Sort Execution Times

|  |  |
| --- | --- |
| Size of Dataset | sortHeap Execution Time |
| 16 | 3200 |
| 64 | 9300 |
| 256 | 45100 |
| 1024 | 213800 |
| 4096 | 1331400 |

Table Heap Sort Execution Times

|  |  |  |
| --- | --- | --- |
| Size of Dataset | sortBasicQuick Execution Time | sortThreePartition Execution Time |
| 16 | 1700 | 1700 |
| 64 | 6500 | 8100 |
| 256 | 31100 | 35500 |
| 1024 | 139600 | 232500 |
| 4096 | 832300 | 963300 |

Table Quick Sort Execution Times

# Graphs of Sorting Algorithm Execution Times

Table Execution time of insertion sort in a normal scale graph

Table Execution time of insertion sort in a log-log graph

Table Execution time of merge sort in a normal scale graph

Table Execution time of merge sort in a log-log graph

Table Execution time of quick sort in a normal scale graph

Table Execution time of quick sort in a log-log graph

Table Execution time of heap sort in a normal scale graph

Table Execution time of heap sort in a log-log graph

# Hypothesis

## Insertion Sort

Based on the graphs of the execution times for each version of insertion sort, my hypothesis is that these algorithms all occupy O(n2) time. This is due to the fact that they appear to have a parabolic trend that is proportional to a quadratic equation.

## Merge Sort

Although the execution times for merge sort are somewhat confusing in the normal scale graph, the log-log graph clearly shows a trend. Based on this trend, I hypothesize that the merge sort algorithms run in O(n log n).

## Quick Sort

The execution times trend for quick sort is very similar to merge sort. The log-log graph has a clear trend that leads me to hypothesize that the quick sort algorithms run in O(n log n) time.

## Heap Sort

Based on the normal scale graph and the log-log graph of the heap sort execution times, my hypothesis is that heap sort runs in O(n log n). This is due to the similarity to the trend of the other O(n log n) predictions.

# Prediction for Larger Arrays

For all the O(n2 ) sorting algorithms, the projection must be proportional to n^2 although this exact value will depend on the running times of the previous sizes of data. As such, we can extrapolate the new values by roughly laying a quadratic equation to fit the previous data. Such a technique tells us that the new values for insertion sort should on average be approximately 10 times greater from 212 to 212 values. From 212 to 216, it should be approximately 300 times longer.

For the O(n log n) algorithms, we can do a similar technique. This tells us that for the merge sorts, it should be approximately 3 times longer from 212 to 214 and approximately 15 times longer from 212 to 216. For Heap sort, it should also be approximately 3 times longer from 212 to 214 and approximately 15 times longer from 212 to 216. For Quick sort, it should also be approximately 3 times longer from 212 to 214 and approximately 15 times longer from 212 to 216.

# Verification

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Size of Dataset | sortInsert Execution Time | sortComparable Execution Time | sortBinary Execution Time | sortMergeTD Execution Time | sortMergeBU Execution Time | sortHeap Execution Time | sortBasicQuick Execution Time | sortThreePartition Execution Time |
| 16 | 800 | 28800 | 3100 | 6400 | 3400 | 3200 | 1700 | 1700 |
| 64 | 2700 | 31700 | 12500 | 22300 | 10600 | 9300 | 6500 | 8100 |
| 256 | 77900 | 97600 | 86700 | 95700 | 46700 | 45100 | 31100 | 35500 |
| 1024 | 2116800 | 859200 | 342400 | 117600 | 206900 | 213800 | 139600 | 232500 |
| 4096 | 11705100 | 17785400 | 1126700 | 533400 | 1186700 | 1331400 | 832300 | 963300 |
| 16384 | 89057900 | 314302400 | 10981700 | 2698400 | 2509200 | 3405900 | 1533200 | 3079700 |
| 65536 | 2600554800 | 9258012800 | 154665800 | 13662200 | 16191700 | 18973600 | 8095300 | 47014000 |

After running the tests, the predicted values were validated because they were roughly around where the measured values average out. The values that were predicted were all around the same magnitude as the empirical evidence and thus, the hypotheses hold true. This new table was generated by creating new data with 214 and 216 data points and re-running the sorting algorithms.