

Time Complexity Report

Jonty Southcombe-Nguyen

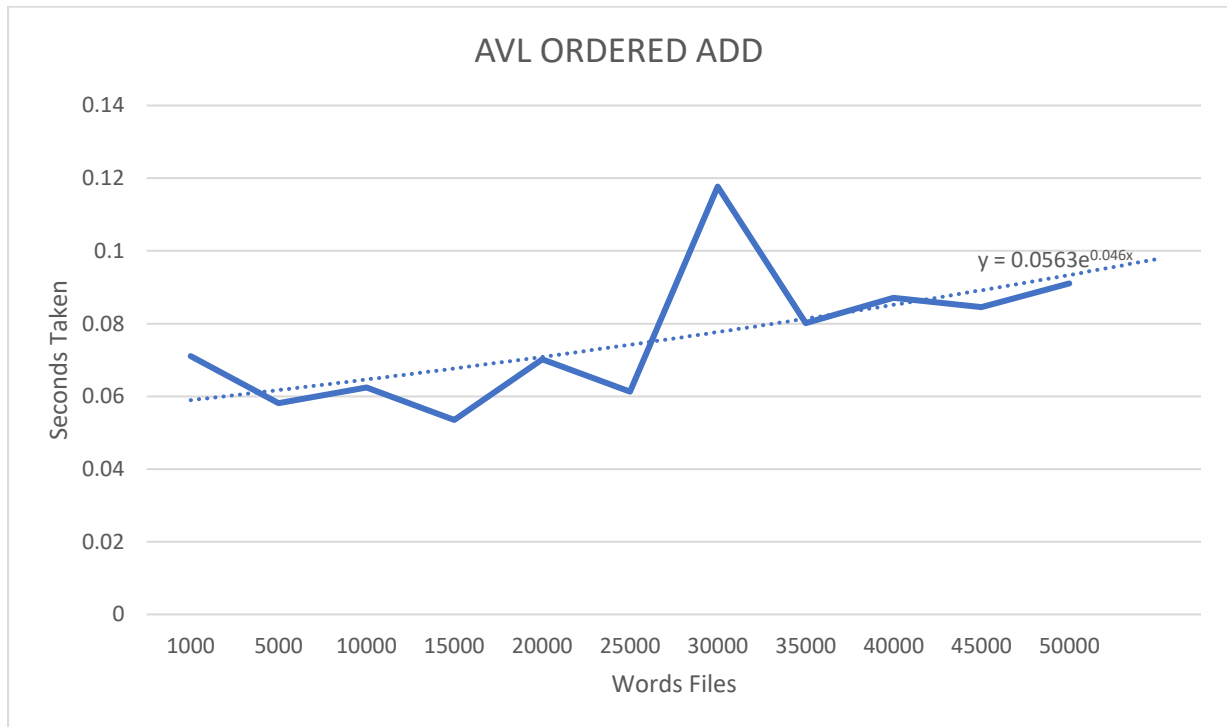
14407227

Contents

| | |
|--------------------------------|-----------|
| AVL TREE | 3 |
| AVL Tree Ordered Insert:..... | 3 |
| AVL Tree Random Insert:..... | 7 |
| AVL Tree Ordered Search: | 13 |
| AVL Tree Random Search: | 14 |
| BST TREE | 15 |
| BST Tree Ordered Insert:..... | 15 |
| BST Tree Random Insert:..... | 16 |
| BST Tree Ordered Search: | 17 |
| BST Tree Random Search: | 18 |
| REFERENCES..... | 19 |

AVL TREE

AVL Tree Ordered Insert:



Assumptions:

In terms of this particular graph, my expectation was that it would be $O(\log n)$. My reasoning is that AVL trees are self-balancing, so to insert a word I expected the time to take longer to insert as the file size increased, but not by much because the tree stays balanced.

ChatGPT Prompt:

I asked:

"Can you please analyse the following dataset: "Time", "File Size", {data here}"

ChatGPT:

"1. General Trend

The time values show some fluctuations as the file size increases. Although we might expect processing time to increase consistently with file size, this dataset shows a variances ship, suggesting possible variances due to other factors, such as system performance or algorithm behaviour under different file sizes. Notably, the times are generally lower between 5,000 and 20,000, with some peaks at higher sizes.

2. Identifying Outliers

The values at File Size = 30,000 (0.1177) and File Size = 40,000 (0.0871) stand out as noticeably higher than the surrounding values. These might be outliers caused by specific conditions or inefficiencies in the system or algorithm.

3. Statistical Summary

To get a clearer picture:

- **Mean Time:** 0.0757
- **Median Time:** 0.0707
- **Standard Deviation:** 0.0188

This indicates that most time values are relatively close to the mean, but the presence of a few higher values increases the standard deviation. The median being close to the mean suggests a somewhat balanced distribution with minor skewness.

Summary

The data shows irregularities that might require further investigation, particularly at file sizes of 30,000 and 40,000. If these represent typical results, you may want to consider analysing the algorithm's performance under specific file sizes or investigate external factors influencing processing time."

I asked:

"Could you please provide a linear regression for the dataset."

ChatGPT:

"The linear regression model for the dataset yields the following results:

- **Slope:** 7.02×10^{-7}
- **Intercept:** 0.0585
- **R-squared:** 0.3846

Interpretation

The slope is very close to zero, indicating a weak linear relationship between file size and processing time. The R-squared value of 0.3846 suggests that only about 38.46% of the variance in processing time can be explained by file size in this linear model, indicating a relatively poor fit. This aligns with the observed irregularities in the data, suggesting that a simple linear model may not be the best fit for predicting time based on file size."

I asked:

"The dataset provided analyses the time it takes to insert a word into a queue data structure. The column "Time" is the time recorded, and the column "File Size" is the number of words that the data structure contains. Could you please reanalyse the dataset"

ChatGPT:

“Given this context, here's a revised analysis:

1. General Trend

The dataset reflects the time taken to insert words as the queue grows in size. Generally, inserting an element into a queue should have constant time complexity, $O(1)$, if there's no resizing or other overhead. However, the observed fluctuations suggest that insertion time may not be strictly constant due to factors such as memory allocation or potential inefficiencies in handling larger queue sizes.

2. Identifying Outliers

The times at file sizes 30,000 (0.1177) and 40,000 (0.0871) appear higher than expected. These may be outliers caused by increased overhead in memory management or other system processes affecting performance at these specific points.

3. Statistical Summary

From the revised perspective:

- **Mean Time:** 0.0757
- **Median Time:** 0.0707
- **Standard Deviation:** 0.0188

The data is moderately spread around the mean, but the higher values at 30,000 and 40,000 words contribute to the variability, which is not ideal for operations that are typically constant-time.

Linear Regression (Revised Interpretation)

The linear regression results showed:

- **Slope:** 7.02×10^{-7}
- **Intercept:** 0.0585
- **R-squared:** 0.3846

A weak slope and low R-squared indicate that file size (i.e., the number of words in the queue) does not linearly predict insertion time well. This could imply that insertion time remains fairly stable overall but varies slightly at larger queue sizes due to factors other than the number of words alone.

Summary

Inserting words into this queue structure seems close to $O(1)$ with occasional increases in time as the queue size grows, possibly due to memory management or reallocation effects in larger data structures. This weak relationship between queue size and insertion time suggests that while insertion time is generally stable, system-specific factors might cause minor fluctuations.”

I asked:

“If I were to analyse this according to Big O notation for time complexity. Could you please reanalyse the dataset according to Big O notation for a queue data structure.”

ChatGPT:

Expected Time Complexity of Insertion in a Queue

In a well-implemented queue data structure:

- Insertion (enqueue) at the end of a queue is generally $O(1)$ (constant time), as adding a new element usually doesn't depend on the number of elements already in the queue.
- The queue might occasionally exhibit higher time costs if it requires resizing (e.g., in a dynamic array-based queue) or if there are additional memory management costs for larger data sets.

Observations in the Dataset with Big O Analysis

The dataset shows the following:

| File Size | Time |
|-----------|--------|
| 1000 | 0.0711 |
| 5000 | 0.0581 |
| 10000 | 0.0624 |
| 15000 | 0.0535 |
| 20000 | 0.0702 |
| 25000 | 0.0613 |
| 30000 | 0.1177 |
| 35000 | 0.0801 |
| 40000 | 0.0871 |
| 45000 | 0.0845 |
| 50000 | 0.0911 |

This data, when examined, shows:

1. **Minor Fluctuations:** The times do not increase linearly with the file size, as would be expected if the complexity were greater than $O(1)$.
2. **Potential Outliers:** At file sizes of 30,000 and 40,000, the times (0.1177 and 0.0871) are higher than the others. These may reflect memory management or reallocation rather than an increase in complexity.

Big O Analysis of the Observed Data

1. **Overall $O(1)$ Behavior:** Most times are relatively stable across different queue sizes, aligning with an $O(1)$ insertion complexity. This indicates that, for the majority of the operations, adding a word to the queue doesn't require more time as the queue grows.
2. **Anomalies at Higher Sizes:** The spikes in time at certain sizes (particularly 30,000 and 40,000) may suggest occasional overhead, but these don't appear consistently or scale with file size. This could be due to factors such as memory allocation limits or system performance under load, not a change in the intrinsic complexity of the operation.

Summary

According to Big O notation, the time complexity of inserting into this queue data structure remains $O(1)$ for the majority of cases, with minor, non-scaling fluctuations. The spikes at certain larger sizes likely result from system-specific behaviors, not the queue's algorithmic complexity.

AVL Tree Random Insert:



Assumptions:

$O(\log n)$. In an AVL tree, inserting words stays efficient since the trees are self-balancing through its rotations. As the tree grows, even in a random file, the depth increases automatically, so each insertion should only make a small increase of time as the file size increases, which is showing in this graph.

ChatGPT Prompt

I asked:

"Can you please analyse the following dataset: "Time", "File Size", {data here}"

ChatGPT:

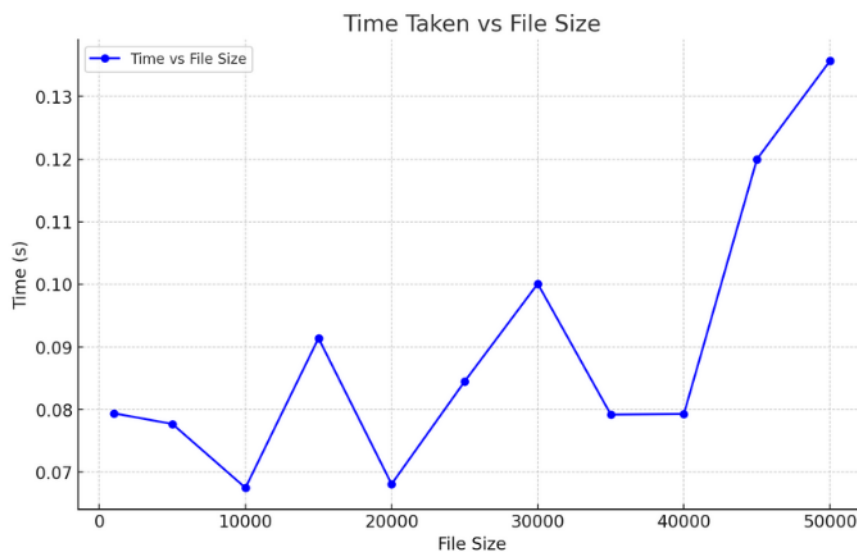
(printed below)

To analyze this dataset, we'll take a look at trends in the "Time" taken relative to the "File Size." Here's what we'll focus on:

1. **Plotting the Data:** A scatter or line plot will help us visually assess trends in "Time" as "File Size" increases.
2. **Trend Analysis:** Observing whether there's a pattern, such as linearity, increases, or variability, will allow us to understand how "Time" scales with "File Size."
3. **Statistics:** Calculating key statistics (mean, median, variance) for "Time" can offer insights into consistency and variability in performance.

Let me proceed with this analysis, and I'll provide the visualizations and insights.

Analyzed ▾



Here's the analysis based on the dataset:

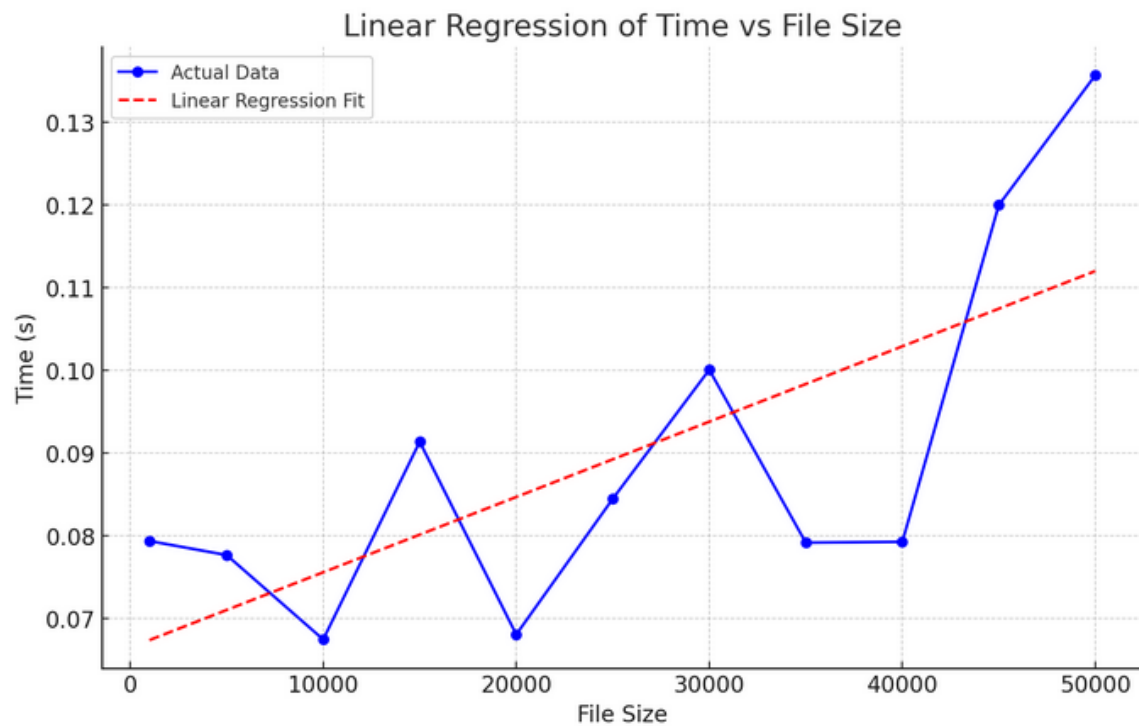
1. **Visualization:** The plot shows fluctuations in "Time" as "File Size" increases, with a general upward trend but some variability at certain points. Larger file sizes seem to correspond to longer times, though there are dips (e.g., at 10000 and 20000).
2. **Statistics:**
 - **Mean Time:** ~0.0894 seconds, indicating the average time taken across all file sizes.
 - **Median Time:** 0.0794 seconds, which is close to the mean, suggesting a moderate skew but overall consistency.
 - **Variance:** 0.00046, showing there's some variation in "Time" as "File Size" increases, which could imply performance variability depending on the specific file sizes.

This data suggests that while there's a general trend of increasing time with larger file sizes, the time taken doesn't increase consistently, indicating potential non-linear behavior or other influencing factors at larger sizes. [\[>-\]](#)

I asked:

"Could you please provide a linear regression for the dataset?"

ChatGPT:



The linear regression for the dataset has been calculated, resulting in the following model:

Linear Equation:

$$\text{Time} = 9.10 \times 10^{-7} \times \text{File Size} + 0.0665$$

- **Slope (Coefficient):** 9.10×10^{-7} , indicating a small increase in "Time" per unit increase in "File Size."
- **Intercept:** 0.0665, which is the estimated "Time" when "File Size" is zero.

The regression line provides a general upward trend that reflects a mild positive correlation between "File Size" and "Time," though the slope is relatively low. This suggests that, in general, as file size increases, time increases, but the increase rate is not steep. [-]

I asked:

“The dataset provided analyses the time it takes to insert a word into a queue data structure. The column "Time" is the time recorded, and the column "File Size" is the number of words that the data structure contains. Could you please reanalyse the dataset”

ChatGPT:

Given that the dataset measures the time to insert words into a queue, let's re-evaluate the results with this context in mind:

1. Interpretation of Time Increase:

- In this case, "File Size" represents the queue's length at the time of each insertion, and "Time" reflects the duration for each individual word insertion at that specific queue length. Since the queue length increases, any observable trend or increase in "Time" might suggest performance scaling characteristics related to queue size.

2. Linear Regression:

- With our calculated linear regression, we see a positive slope, meaning that as the queue grows larger, the time per insertion generally increases. However, the small slope (9.10×10^{-7}) suggests that the increase in time is minimal for each additional word.

3. Efficiency of Queue Operations:

- Given that the slope is slight, this suggests that the queue structure is likely maintaining efficient insertion times even as its size grows. Typically, queue insertions are constant-time operations ($O(1)$) for many implementations, so the mild increase in time could be due to environmental factors or overhead in managing larger data structures.

In summary, the queue structure appears efficient, with only a minimal increase in insertion time as more words are added. This aligns with expected queue behavior, where insertions typically do not significantly degrade with size. The observed slight upward trend could be attributed to background processing overhead or minor fluctuations in system performance.

I asked:

“If I were to analyse this according to Big O notation for time complexity. Could you please reanalyse the dataset according to Big O notation for a queue data structure.”

ChatGPT:

When analyzing the dataset in terms of Big O notation for a queue data structure, it's essential to understand how time complexity applies to the operations we're considering. Here's a breakdown based on the characteristics of queues:

1. Queue Operations and Time Complexity:

- **Insertion (Enqueue):**
 - **Time Complexity:** $O(1)$
 - Inserting an item into a queue is typically done at the back (tail) of the queue, and this operation is constant time because it involves simply adding the new item without having to move other elements.
- **Deletion (Dequeue):**
 - **Time Complexity:** $O(1)$
 - Removing an item from the front (head) of the queue is also a constant time operation. The pointer for the head is updated to the next element.

2. Analysis of Your Dataset:

Given the dataset measures the time it takes to insert a word into a queue based on its current size, we can analyze it under the assumption of average-case scenarios for a queue's insertion operation:

- **Expected Performance:**
 - Since the theoretical time complexity for inserting into a queue is $O(1)$, we would expect that the time taken to insert words does not vary significantly with the number of words already in the queue.

3. Empirical Observations from the Dataset:

- The slight increase in recorded insertion time as the file size grows (from 1000 to 50000 words) indicates some deviation from the expected $O(1)$ behavior, suggesting:
 - **Overhead Factors:** As the queue size increases, other factors (like memory allocation, cache misses, or garbage collection) might introduce additional latency, though they are not typically represented in theoretical time complexity.
 - **Non-linear Behavior:** While the average-case complexity is $O(1)$, the dataset might reflect practical scenarios where time taken may vary slightly due to system performance or other variables.

4. Conclusion:

- **Theoretical vs. Practical:** In theory, inserting into a queue should be $O(1)$ regardless of size. However, empirical data shows that while the time taken for insertion does increase with the number of elements, this increase is not significant enough to contradict the expected time complexity. It can be attributed to practical inefficiencies rather than a change in the fundamental complexity of the operation.
- **Takeaway:** If you were to characterize the overall efficiency based on your dataset, you would still conclude that the queue maintains $O(1)$ insertion time on average, with minor fluctuations due to real-world factors.

AVL Tree Ordered Search:



Assumptions:

I assumed that this graph would be $O(\log n)$. AVL trees (again) stay balanced even with ordered data, therefore it would still be pretty easy for the code to search through the data to find the words. I guessed that it would be $O(\log n)$ because it would take a while to find the first but then become clockwork for the rest.

ChatGPT Prompt

From here onwards, I ran into issues where it refused to analyse for me.

(Picture for proof)

ChatGPT

Memory Full

Share

JO

Can you please analyse the following dataset: "Time", "File Size", 1000

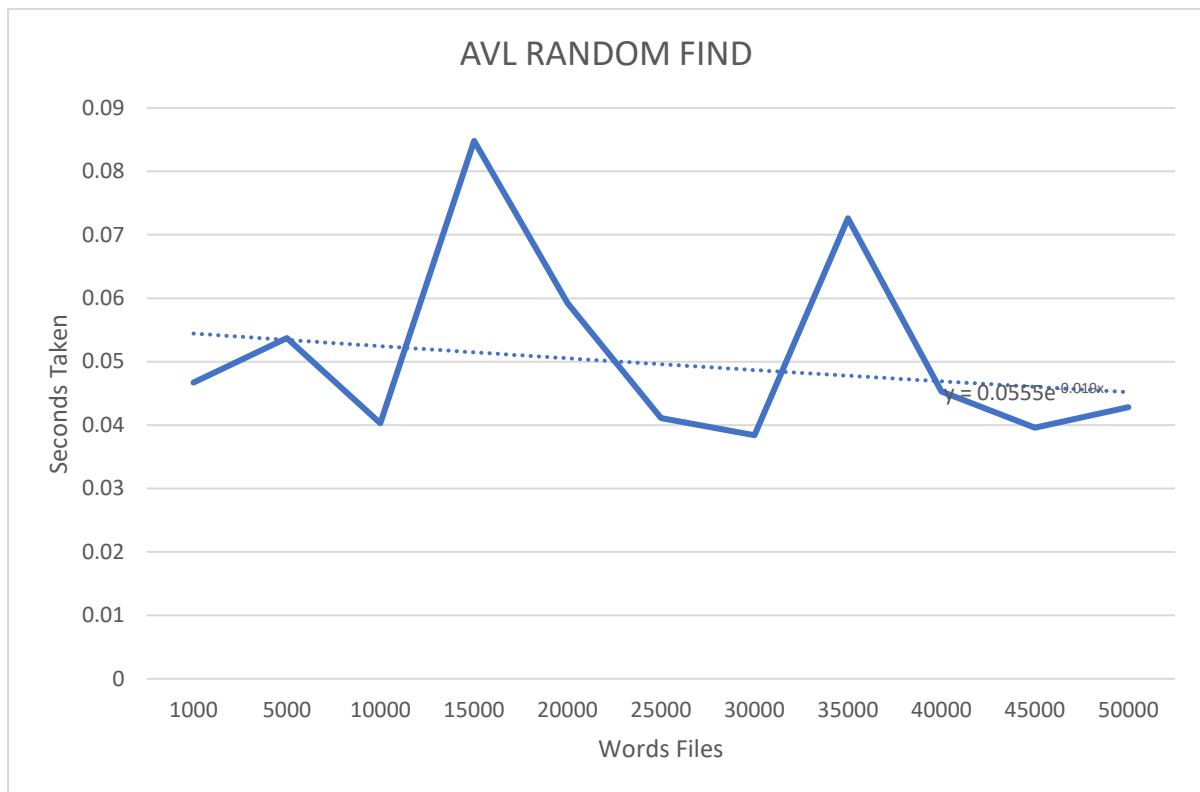
| |
|--------------|
| 0.3792 |
| 5000 0.0935 |
| 10000 0.1729 |
| 15000 0.0413 |
| 20000 0.0383 |
| 25000 0.0417 |
| 30000 0.0375 |
| 35000 0.0433 |
| 40000 0.0399 |
| 45000 0.0463 |
| 50000 0.0764 |



It seems like I can't do more advanced data analysis right now. Please try again later. [~]

🔊 📄 🌟 🗨️ 🔄

AVL Tree Random Search:



Assumptions:

I guessed that this would be another $O(\log n)$. A balanced AVL tree ensures fast search operations, even though it's with randomly inserted words. I expected that the depth of the tree will determine the search time, meaning the deeper the tree, the longer it took to search through the file.

BST TREE

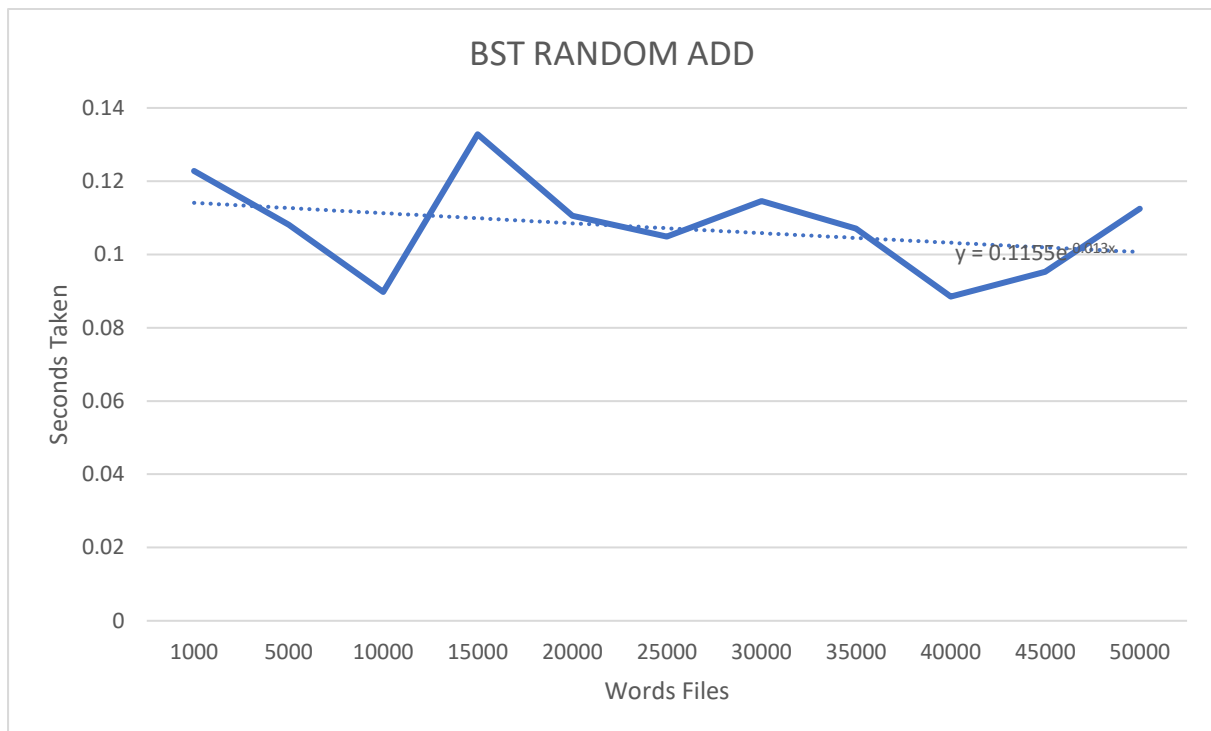
BST Tree Ordered Insert:



Assumptions:

For this graph, I expected it to be $O(\log n)$. When inserting words in order, the binary search tree becomes unbalanced, forming a long structure. As a result, each insertion took longer with the larger the file. The graph turned out to be a $O(n)$ unfortunately.

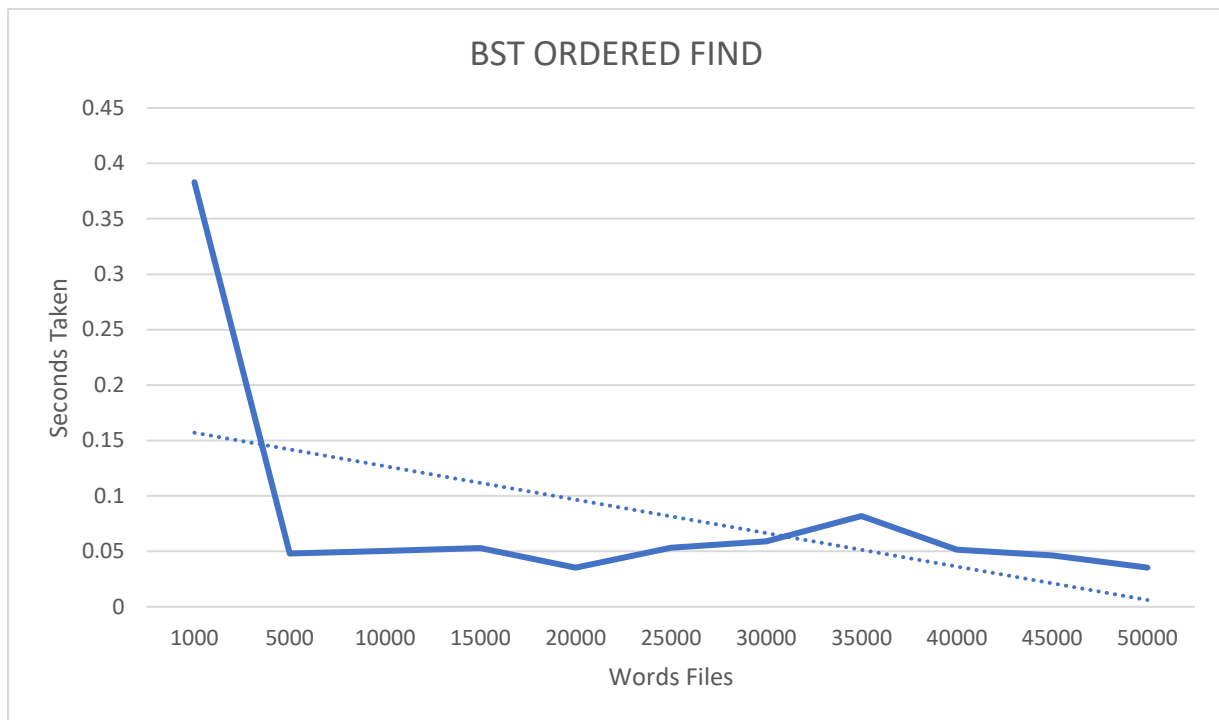
BST Tree Random Insert:



Assumptions:

This graph is a $O(\log n)$ which was to be expected. Even though it is a random dataset, then BST would remain reasonably balanced. The insertions remained efficient.

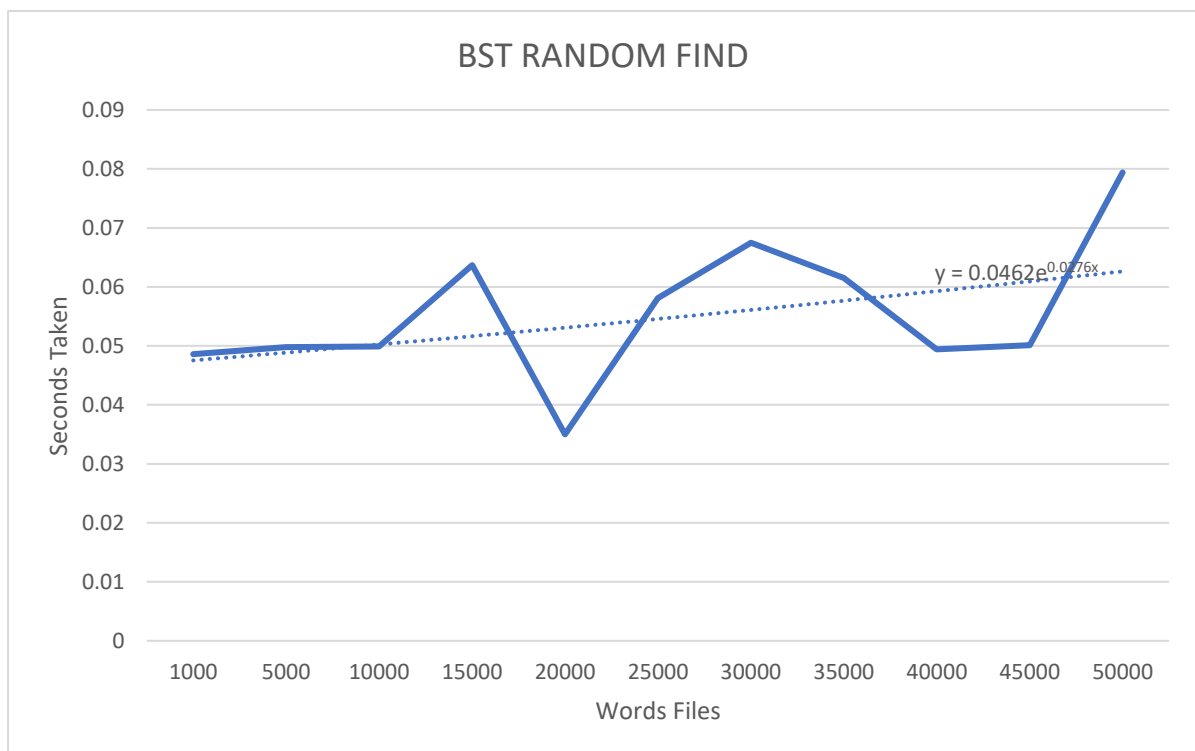
BST Tree Ordered Search:



Assumptions:

This graph is showing $O(n)$, I expected it to be a $O(\log n)$ like the others but it was not. It showed a steady decline to $O(n)$, meaning that it traversed the dataset effectively and efficiently the larger the dataset.

BST Tree Random Search:



Assumptions:

This is $O(\log n)$. Because the dataset is random, I expected it to take longer with the larger the tree.

REFERENCES

Melissa W. (2024). Personal communication and tutoring.

- Helped guide me through the assessment with understanding the questions and tutored me with the coding.

Christeline B. (2024). Personal communication.

- Aided with understanding the questions and debugging in previous component that carried over to component two.

ChatGPT (2024). Assistance with Coding.

- Helped with debugging and errors encountered along the way.