Edward Rees 20510170
*College of Arts and Science*
*University of San Francisco*

**November 17, 2019**

**ASSIGNMENT 1 — Analysis**

**Average Big O of Trie Functions**

m = length of string

| Function | Worst | Average |
|---|---|---|
| Search | O($m$) | O($m$) |
| Insert | O($m$) | O($m$) |
| Suggestive Add | O($m^2$) | O($m^2$) |
| Suggestive Remove | O($m^2$) | O($m^2$) |
| Suggestive Substitute | O($m^2$) | O($m^2$) |
| Suggestive Adjacent | O($m^2$) | O($m^2$) |

**Average Big O of Search Tree Functions**

n = number of words to search through
m = length of string

| Function | Worst | Average |
|---|---|---|
| Search | O($n$) | O($logn$) |
| Insert | O($n$) | O($logn$) |
| Suggestive Add | O($n * m$) | O($m * logn$) |
| Suggestive Remove | O($n * m$) | O($m * logn$) |
| Suggestive Substitute | O($n * m$) | O($m * logn$) |
| Suggestive Adjacent | O($n * m$) | O($m * logn$) |

## Faster Data Structure

As shown above, it depends on the value of m and n to really dictate what the best algorithm is. If $n$ is greater than $m$, meaning the number of elements is greater than the longest word, then a trie would be faster. However, if the value of $n$ is less than $m$, or the number of elements is less than the longest word, then a search tree would be faster.

None of the suggestive functions are O($m$) as they all use the search function itself, which, for the trie, is already O($m$). Therefore the suggestive functions for the trie is a factor of O($m$) or O($m^2$).

For the search tree itself, since the search function itself is O($n$) at worst, or O($lgn$) on average, the suggestive functions would be a factor of the search function times $m$. Meaning, on average, the suggestive functions would be O($m * logn$) and at worst, would be O($n * m$). Again, if $n > m$, a trie is faster. If $n < m$, a search tree is faster.

## Extra Credit Information

The code will naturally try to read the $english.0$ file from GitHub, if it can't, it will read from the local file. If the local file doesn't exist, it will create a file with the same name and try to work off of that.

If the name of the files inputted don't exist ($input.txt, output.txt$) then the files will automatically be created and used.

*Submitted by Edward Rees 20510170 on November 17, 2019.*