Wordstat Documentation

Matt Klein

February 13, 2013

1 Intro and About

This program, `wordstat`, is meant to count words passed into the program via a file. Typing -h will display a help file, and when typing in an invalid file name. The file names are **case-insensitive**, meaning if a file exists named `TEST.txt` typing in `test.txt` will read `TEST.txt`. will display that the file does not exist. The `makefile` provided will create wordstat from `wordstat.c` and `wordstat.h` using the GCC compiler, with the `-ansi -pedantic -Wall` flags. At the very end of the program, the program decides to use `fclose` function to clear memory along with the `free` function.

2 Data Structures

This program uses a Binary Search Tree (BST) along with a Linked List in order for a **reasonable** worst case time. Despite not being perfect, it is not the worst possible run time.

3 Run Time Analysis

Because we are using a BST, the worst case of each method is the following:

- `Insertion -` $O(n^2)$

  This is only $O(n^2)$ when we insert **completely** to one side. Otherwise, we will always get $O(logn)$. This can be improved by using a self-balancing tree, rather than one that does not.

- `Print -` $O(n)$

  This is always $O(n)$ as we are going through a tree with n items. Sometimes can be $O(n+l)$ where l is the size of the linked list.

- `Clear` - $O(n+l)$

  This is always $O(n+l)$ as it is the size of the tree. Every node and item in the linked list is cleared.

- `Help` - $O(1)$

  This is always instant because we just use `printf(...))` statements.

- `Substring` - $O(1)$

  This is always instant because the start index and end index are passed into the function.

- `Find Letter` - $O(n)$

  Find letter finds the first letter, and we must assume that k is the length of a given word. Assuming that $k = n$ (the entire document), the worst case is the index is at the last possible spot. Usually, this function is $O(k)$.

- `Find (Linked List)` - $O(l)$

  Finding an item in the linked list takes a max of l time, where l is the size of the linked list at the given node.

- `Add (Linked List)` - $O(1)$

  Adding to the end of the linked list gives us a max of **constant** time, as the previous function, `Find (Linked List))` will always go to the end of the linked list. Add just tags on an extra node.

4 Space Analysis

The program can only read up to a `BUFFER_SIZE` of 100. However, the program continues to read new lines and will continue to update. At the end, `fclose` and `free` are used to free memory from the file and the BST.