

PointerSorter Readme

Program summary:

This program is a simple C tokenizer, where it takes one input string and parses it into words. The delimiter characters are simply any non-letter character, such as “!” or a space.

Implementation:

The program uses two indices to parse the words, one to iterate through each character in the input string and one character to represent each non-letter character found in the string. The algorithm simply iterates until a non-letter character is reached, then forms a substring from the last non-letter index to the current one, and then stores that word into a data structure of choice.

Functions:

- **create_node:**
 - Creates a binary tree node, with a key and left and right children.
- **insert_word:**
 - Inserts a word into the binary tree by creating a node with the word as its key and then adding it as a standard binary tree insert operation.
- **in_order_traversal:**
 - Performs a standard in-order traversal of the binary tree to print out the tree’s words in sorted (alphabetical ASCII) order.
- **destroy_tree:**
 - Frees the memory allocated dynamically to create the tree. Memory is freed using a post-order traversal.
- **get_substring:**
 - Returns a substring of an input string given start and end indices.
- **parse_input:**
 - Parses an input string into tokens, with delimiter characters being any non-letter character, then inserts each token into the binary tree.

Data structure(s):

To store the parsed words from the input string, we chose to use a binary tree, with each node containing a word. The reason we chose to use a tree was due to the general efficiency offered by binary trees; insertion is worst case $O(n^2)$ but $O(\log n)$ on average, and it has the benefit of being already sorted. Rather than inserting all words and sorting after, one can simply perform an in-order traversal of the binary tree to get all the words in sorted order. The sorting order we chose was the sorting order determined by the C string library’s `strcmp` function, which is alphabetical ASCII order.