Text Search: Algorithm Report

Alvin Dang

Department of Computer Science, San Jose State University

CS 146: Data Structures and Algorithms

Vidya Rangasayee

September 20, 2023

Describe your algorithm:

My algorithm was designed to read a text file and search for an array of names.

It works by reading each line of the text file and parsing it in search of the desired names. When a name is found, it then outputs the line the name was found on, the starting character position, and the name found.

My algorithm uses the 'BufferedReader' class. I chose this specific class as it buffers the input data, which I believe is a very efficient way to read large bodies of text.

In the event that a name is spread across multiple lines, the algorithm merges the last word of the previous line with the current line. Furthermore, in the event that a name follows a punctuation mark, the algorithm adjusts the position of the name to correctly suit its logical position.

Overall, my algorithm has a time complexity of O(nm), where n is the number of lines in the given text file and m is the length of the longest desired name.

How well will the algorithm scale with respect to the length of the text and the number and lengths of the names to search?

My algorithm scales linearly with respect to the number of lines in the text file. Meaning as the text file becomes larger, the algorithm would also take a longer amount of time (ms) to run. Nonetheless, I believe my algorithm is equipped to handle larger text files. Given there is enough available memory to store the file.

My algorithm should continue to scale linearly even when considering how well it would scale with respect to the number and lengths of the names to search. Meaning just like before, the more names and the longer they are, then the longer the algorithm will take to run.

With all that said, a linearly scaling algorithm is one that can likely be improved upon. I have no doubt that optimization of the algorithm or the use of a completely different algorithm with better time complexity would be required for excessively large text files.