

Indexer : Calin Gilan & Saif Saiyad

This indexer counts the amount of occurrences of words in either one or multiple text files. If asked to, it will recursively scan directories looking for text files and add their occurrences to a list in memory. This list is then printed out in a JSON compatible text file.

Efficiency and Design

The program stores each word in a linked list. Each word has its own node, and each node has a pointer to the head of a node that contains a list of all the places that word occurs, and the amount of times it does.

Word Node

Word Storage

Head of node that contains linked list of files and repetitions

Pointer to next Node

File Node

File name

Amount of times the word is repeated

Pointer to the next node

The program starts by recursively going through each file and directory, and then checking if each file is a text file. This is an $O(n)$ operation, depending on how many files and directories there are to go through. It then stores the locations of the text files in a linked list.

The program then goes through the linked list and opens all those files and makes each word a node, with all the properties described earlier.

Doing it this way allowed for ease of implementation, modularity and scalability, at the cost of efficiency and space.

Space wise, this means that worst case scenario is $\text{Size of Text File} + (\text{Words} * \text{Size of Word Struct}) + (\text{Words} * \text{Amount of Files} * \text{Size of File Node})$.

This assumes all unique words that only appear once.

Efficiency wise is , $O(n*m)$ where n is the amount of unique words and M is the amount of files.