Srihari Chekuri & Ajay Srivastava

READ ME

Similar to assignment one, we made an algorithm to search for words in a file and sort them, but we used a different data structure to make the searching faster. We used a trie tree to store the words from each folder and then sort them. The trie tree nodes have an array size of the alphabets (created only when necessary) and each node will have several attributes along with a fileNodePointer (Gives count for token in a given folder). The files are first sorted alphabetically, and then we parse through the words (create tokens) in each file and sort them alphabetically. Then we wrote another algorithm to sort the fileNodePointers for each token. The tokens are first arranged based on the number of occurrences and when the token has the same number of occurrences in another file, then it sorts the fileNodepointers alphabetically. The search time using a regular linked list has a worst case of the size of the linked list, whereas a search performed in trie tree has a worst case time of the size of the token. The search time for a trie tree is nlog(n). Thank You, have a nice day.