

Functions `coord`, `trieEdge` and `idTrieEdge` has a time complexity of $O(1)$ as all operations are constant and space complexity of $O(1)$ as size of variables does not scale based on inputs.

Function query

Data is read from the file in $O(NM)$ where N is number of lines and M is length of biggest sentence in the file. Then a trie is created from data obtained from each line that is read. Inside the outer loop that reads lines one at a time, there runs a for loop for the length of last name/ID (depending on which trie is being made). Hence, the tries are created in $O(T)$ where T is the number of characters in ID's and last names.

The query sections of the function loop k and l times respectively for ID and last name at first, when the prefixes are searched for in both tries.

To combine the outputs from both loops, a loop runs maximum of $nk + nl$ times. Hence, time complexity for querying is $O(k + l + nk + nl)$.

Space complexity of the function does not exceed $O(T + M)$. Both tries do not combine to be larger than T , and the file is read one sentence at a time, where M is the largest sentence in the file. No other stored variables exceed either T or M .

Function reverseSubstrings

Every loop inside this function only runs a maximum of K times (number of iterations labelled for each loop in the code) where K is number of characters in the input string. The maximum time complexity for the function is $O(K^2)$ (due to nested loops). Space complexity never exceeds $O(K^2 + P)$ as all suffix arrays and the trie fall under $O(K^2)$, and the output result falls under $O(P)$.