query(filename, id_prefix, last_name_prefix)

Algorithm: Open the file and read through the lines, initialize 2 empty strings one to store the characters the other for temporary storage. Concatenating the lines of the file and then starting a for loop to check the character only between A to Z, a to z and 0 to 9. Then the space is checked and if there is a space it check the column by the column counter, if it's the first column then set the string as id, and if its on the third column set it as word(last_name) and increment the columns in the end. Then if there is a new line, add the word, id and the index to the trie and increment the index to the next record and set the column back to zero. A trie object is created and trie search takes place to find the respective prefix_id and prefix_last_name. Two lists are returned as I have one trie using two different nodes, one for the alphabets and one for the id. Overlaps are searched in the end to find the matching records and the final list is returned.

Creating Tries: Two nodes and one trie was used to add and search the words. The add function had two loops one for the words and the other for the id. The search function had two loops as well, one for the word and one for the id. Two get function were used to, the first get converted the alphabets irrespective of their sensitivity while the second get simply converted the numbers to the index. The search function returns the index of two lists which later on is used to find the overlaps

precondition: word and id
post-condition: list of indexes
Time Complexity: $O(k+l+ n_k + n_l)$ where k is the length of id_prefix, l is the length of last_name_prefix, n_k is the number of records matching the id_prefix and n_l the number of records matching the last_name_prefix

Short Summary:

Preprocessing takes O(NM) time to read the file. When adding the id and last_name to the trie, it takes a total of O(k+l) where k is length of the last_name and l is the length of the id. After adding into the trie, the search operation also takes the same time complexity as adding in to the trie. In the end to find the overlaps it takes a total of $O(n_k + n_l)$ where , n_k is the number of records matching

the id_prefix and n_l the number of records matching the last_name_prefix. So the total time complexity is O(k+l+ n_k + n_l).

Space Complexity: O(T+NM) T is the number of characters in all identification numbers and all last names which are used to make a trie and NM is reading the input.