**Aim:** Implement the simplified Search Engine described for the pages of a small Web site.

**Approach:**

We have implemented a Trie data structure in python to create a search for the input documents (HTML pages).

Trie is an efficient information re***Trie***val data structure. Using Trie, search complexities can be brought to optimal limit (key length).

The first step is to download the HTML files to the input directory from which the program will traverse in order to create a Trie. We use Beautiful Soup to parse the input documents to extract the HTML elements. After extracting the text from HTML, we tokenize the words in the text using NLTK library’s word\_tokenize() function. We sanitize the tokenized words by removing stop words such as articles, prepositions, and pronouns along with removing special symbols i.e. non ascii symbols. The tokenized words are then inserted into the Trie. The program then asks the user for the input word which needs to be searched in the documents. Once the user provides the input, the Trie is traversed to searched for the input word and the result is presented to the user. The websites have been ranked using the counting method, in which it checks for the occurrence of the word in the webpage, and returns the search results in the decreasing order of the word occurrence.

**Trie insertion:**

1. Every character of input key is inserted as an individual Trie node.
2. The child elements in the children array point to next level Trie nodes**.**
3. Key refers to the word that you are inserting or searching in the Trie.

**Trie search:**

1. While searching we only compare the character and move down the tree.
2. If the value field of last node is non-zero then the key exists in Trie.
3. Search will be terminated due to reaching the end of string or due to lack of key nodes in Trie without examining all characters

Insert and search cost **O(k)** where **k** is length of the key.

Memory requirement of the Trie is O(ALPHABET\_SIZE\*k\*N) where N is number of keys in Trie.

**To run the program (example given for Mac Terminal):**

> pip install -r requirements.txt

Go to the directory, for example: > cd Desktop/SearchEngine

> python Search\_Engine.py

If it doesn’t run, it is because of the Mac hidden file- **.DS\_Store**, please remove and run again. To do that:

Go to the input folder, for example: > cd Desktop/SearchEngine/input

> ls – a (should show .DS\_Store)

> rm .DS\_Store