Trie Data Structure

Trie means retrieval.
Trie is sorted tree data structure that stores string.
Has pointers equal to the number of character of the alphabet in each node.

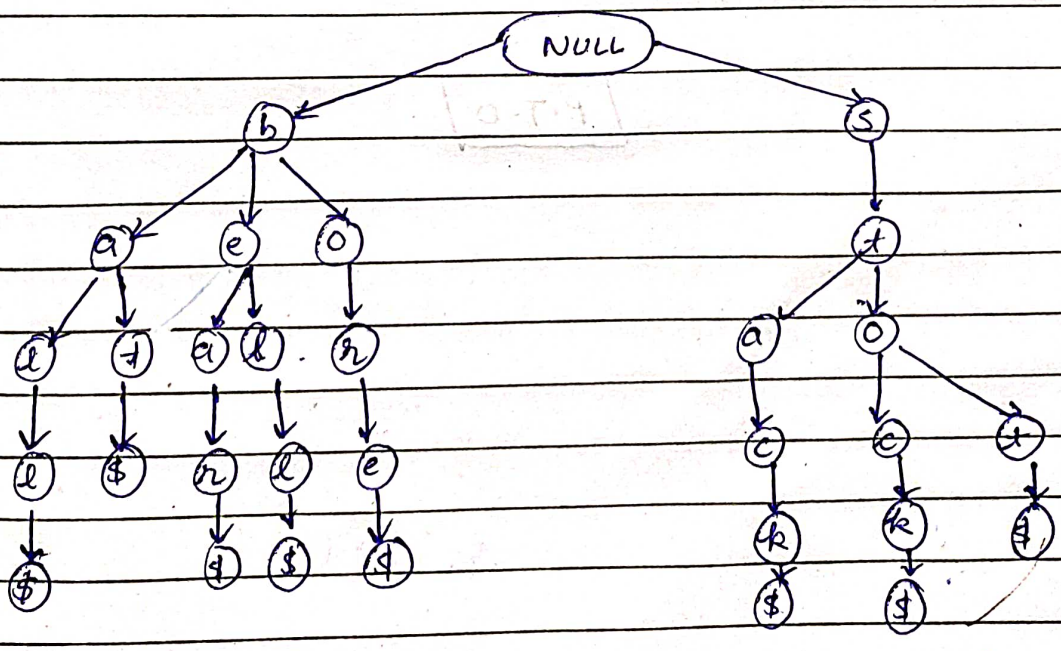
For example :-

If we assume that all the strings are formed from letters 'a' to 'z', each trie node can have maximum 26 points.

Properties of Trie for a set of string :-

The root node always represents the null node.
Each child of nodes is sorted alphabetically.
Each node can have maximum of 26 children (A-Z)
Each node (except the root) can store one letter of the alphabet.

The diagram representing for (i) Bell, (ii) bear, (iii) bore, (iv) bat, (v) ball, (vi) stop, (vii) stock, (viii) steck.



Trie insert and Search :-

→ Insertion and search is same in trie DS.

Remember →

1- Every letter of the input key (word) is inserted as an individual in the trie node. Note that children point to the next level of trie nodes.

2- The key character array acts as an index of children.

3- If the letter is not already represented by the person, we create a new node for the letter and link them together. (node)

4- The depth of Trie corresponds to the length of word. Each level down is like adding one more letter to the word.

Time Complexity :-

Insertion, Deletion and searching :- O(L)
↳ length of key.

P.T.O