Tries

All the search trees are used to store the collection of numerical values but they are not suitable for storing the collection of words or strings. Trie is a data structure which is used to store the collection of strings and makes searching of a pattern in words more easy. The term ***trie*** came from the word **retrieval**. Trie data structure makes retrieval of a string from the collection of strings more easily. Trie is also called as **Prefix Tree** and some times **Digital Tree**. A trie is defined as follows...

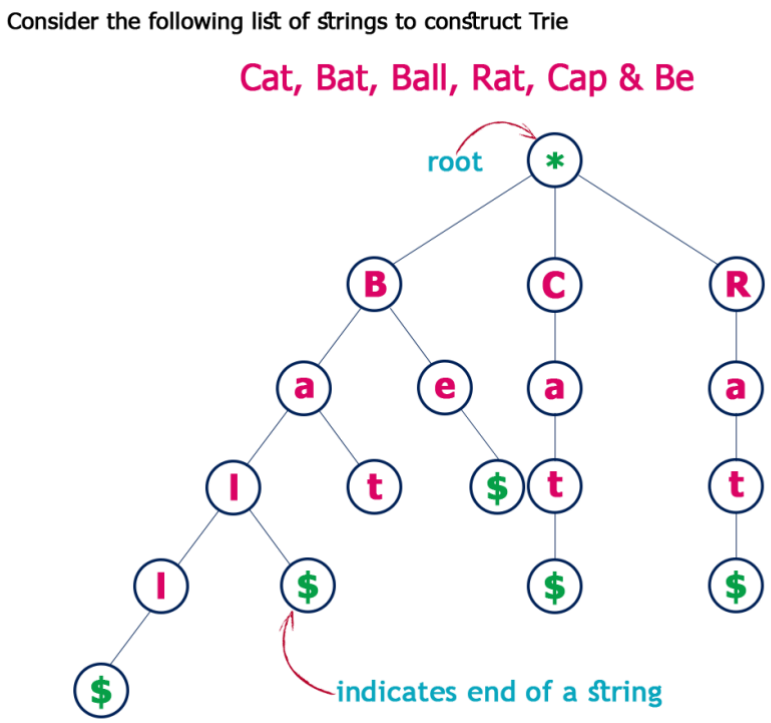
Trie is a tree like data structure used to store collection of strings.

A trie can also be defined as follows...

Trie is an efficient information storage and retrieval data structure.

The trie data structure provides fast pattern matching for string data values. Using trie, we bring the search complexity of a string to the optimal limit. A trie searches a string in **O(m)** time complexity, where **m** is the length of the string.
In trie, every node except the root stores a character value. Every node in trie can have one or a number of children. All the children of a node are alphabetically ordered. If any two strings have a common prefix then they will have the same ancestors.

Example



1. In computer science, a tries is also called digital tree and sometimes radix tree or prefix tree.
2. Tries is a tree based data structure for storing strings in order to support fast pattern matching.
3. Tries are used for information retrieval.
4. Tries is used to store the character in each node not the key.
5. Path from root to node is associated with key.
6. Tries uses character of a key to guide the search process.
7. All the descendants of the node have a common prefix of the string associated with that node.

Types:

I. Standard Tries:

The standard tries for a set of strings S is an ordered tree such that:

- Each node but the root is labeled with a character.

- The children of a node are alphabetically ordered
- The paths from the external nodes to the root yield the strings of S.
- Example: Standard tries for the set of strings S ={ bear, bell, bid, bull, buy, sell, stock, stop} is shown in figure 14.

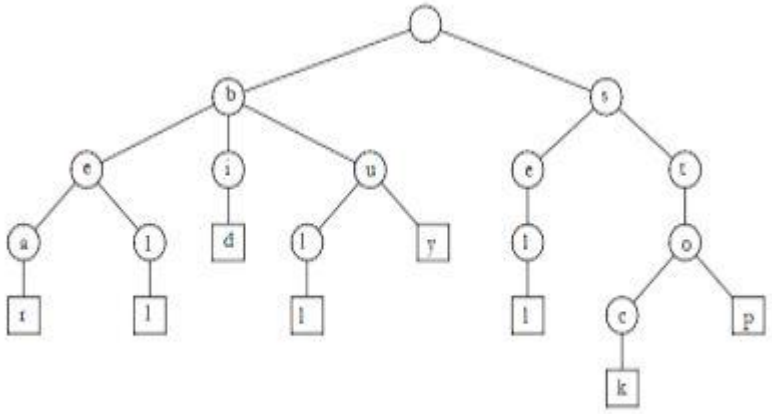


Figure 14

II. Compressed Tries:

- Tries with nodes of degree at least 2.
- Obtained from standard tries by compressing chains of redundant nodes.

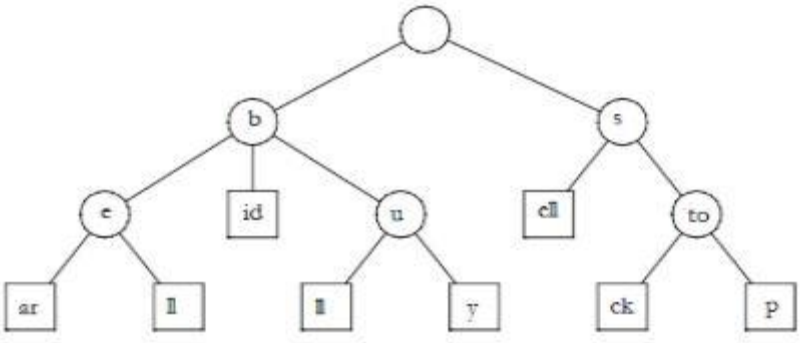


Figure 15

III. Suffix Tries:

- A suffix trie is a compressed trie for all the suffixes of a text.
- The suffix trie for a text X of size n from an alphabet of size d.

- Suffix tries stores all the n(n−1)/2 suffixes of X in O(n) space.
- Suffix tries supports arbitrary pattern matching and prefixes matching queries in O(dm) time, where m is the length of the pattern.
- Suffix tries can be constructed in O(dn) time
- Applications:
- Word matching.
- Prefix matching.

- Example: Minimize example is shown in figure 16.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
M	I	N	I	M	I	Z	E

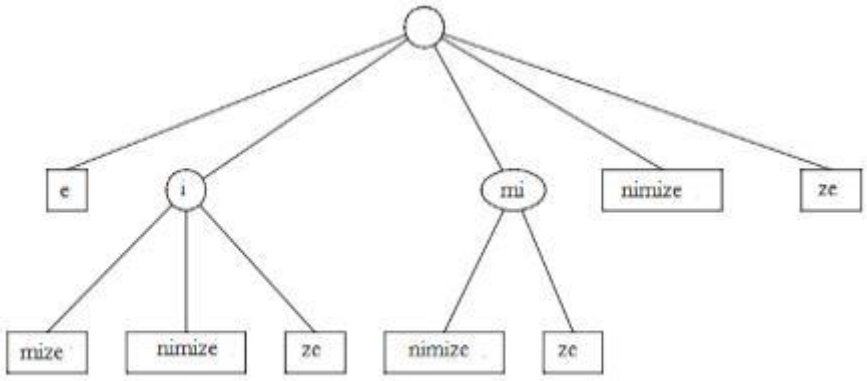


Figure 16