Tries

Trie data structure is a type of ***multi-way tree*** that is used for ***storing different*** strings. Each string consists of characters that are stored in a ***tree-like structure***, i.e., ***Trie data structure***. It is also called a ***radix tree or prefix tree***, or ***digital tree***. Basically, the word "**trie**" comes from the word "re**trie**val", which means ***retrieving*** or getting back something. It is used for various tasks such as s***pell-checking, searching words, auto-completion***, etc.

Properties of Trie data structure:

- ***Trie data structure*** is the form of a ***tree structure*** where each node represents a ***single character*** of a string, and the path covered from root to node represents a ***specific string***.
- The ***trie data structure*** starts from the ***root node***, which is always empty and represents a ***null character***. From the root node, various other branches emerge that hold other characters of a string.
- The end of a string in a ***trie data structure*** is called a ***leaf node***. Each leaf node may contain extra information.

Operations in a Trie data structure:

There are ***three operations*** that can be accomplished in a trie data structure:

1. Insert operation:

This operation is used to ***add*** a ***new node***, i.e., a ***new string***, into the ***Trie***.

2. Search operation:

This operation is used to ***find*** a specific string and check if it exists in the ***Trie data structure***.

3. Delete operation:

This operation is used to ***remove*** a string that is present in the ***Trie data structure***.