TRIES

Tries deal with Strings.

Normally strings are stored as arrays.

The challenge with arrays is:

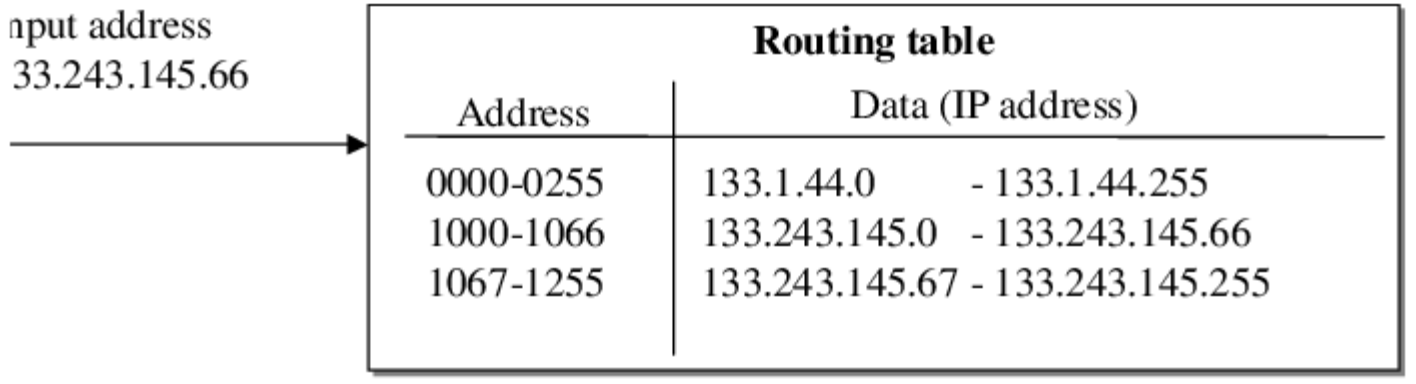
- Large number of strings can’t be handled.
- Search time will be high

Applications:

- Search Engine results optimization.
- Data Analytics.
- Sentimental Analysis. The data structure that is very important for string handling is the Trie data structure that is based on prefix of string.
- DNA Analysis

Other applications:

1. **Auto Complete – Web browers, search engines, editors, email, word processors**
2. **Spell Checkers**
3. **Longest Prefix Matching – Computer networks → Routing**
4. **Browser History**



Tries

- Trie is an efficient information re***Trie***val data structure.
- A trie is basically a tree.
- It is also called as digital tree or prefix tree.
- It represents (or stores) a set of strings.
- Every node of Trie consists of multiple branches. Each branch represents a possible character of keys.
- i.e., At each node, we store one letter.
- We need to mark the last node of every key as end of word node.

TRIES are useful for handling strings. The String processing has a variety of real world applications too, such as:

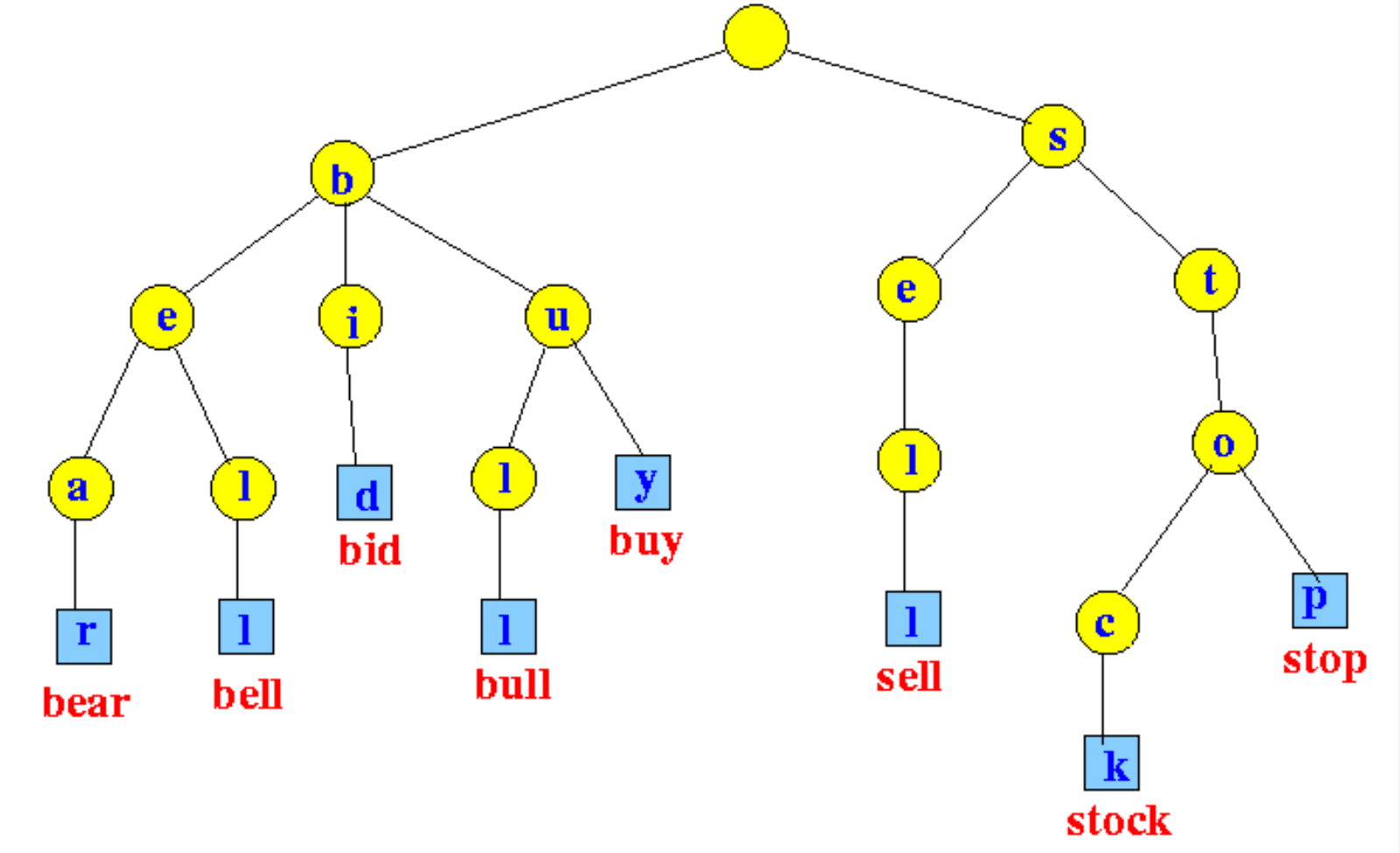
- Search Engines
- DNA Analysis
- Data Analytics

Types of Tries:

1. Standard Trie
2. Compressed Trie
3. Suffix Trie

Eg. S = bear, bell, bid, bull, buy, sell, stock, stop

A trie is given below:



Running time for operations:

- A standard trie uses O(W) space.
- Operations find, insert, delete take O(dm) time,

where:
- W = total size of strings in S
- m = size of string involved in operation
- d = alphabet size

Eg: in Contacts App of Mobile:

Srinivas
Srinu
Sridevi
Sridhar
Srikar
Srimukhi

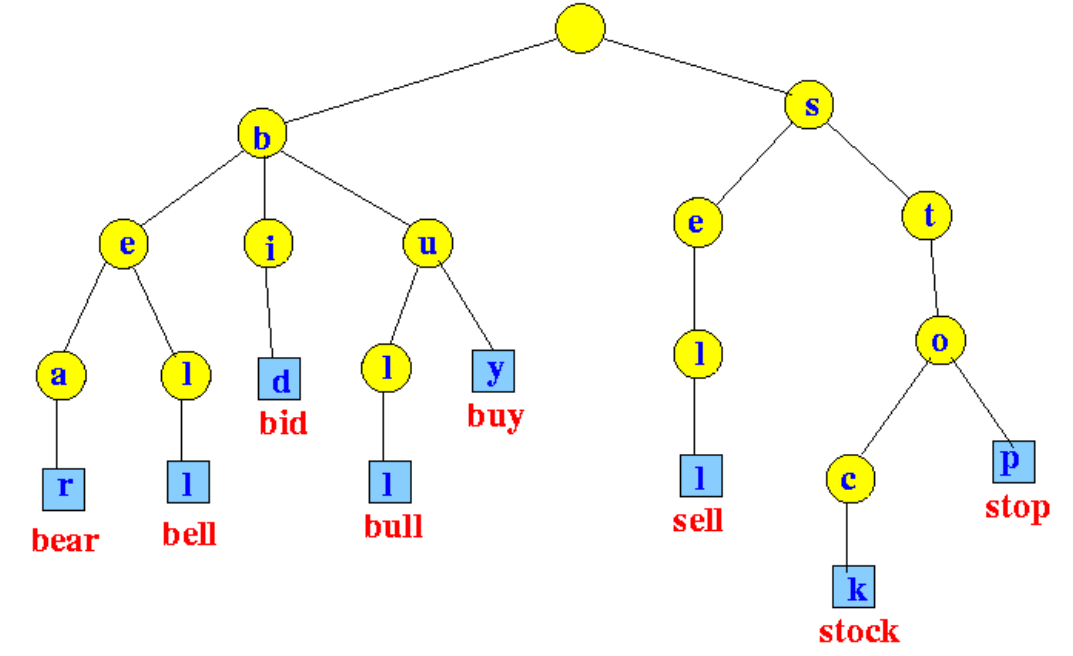
Compressed Trie:

- A compressed trie is similar to a Standard trie in a more compact fashion.

- The idea of the compressed trie tree is to convert long chains of single-child edges to one single edge.

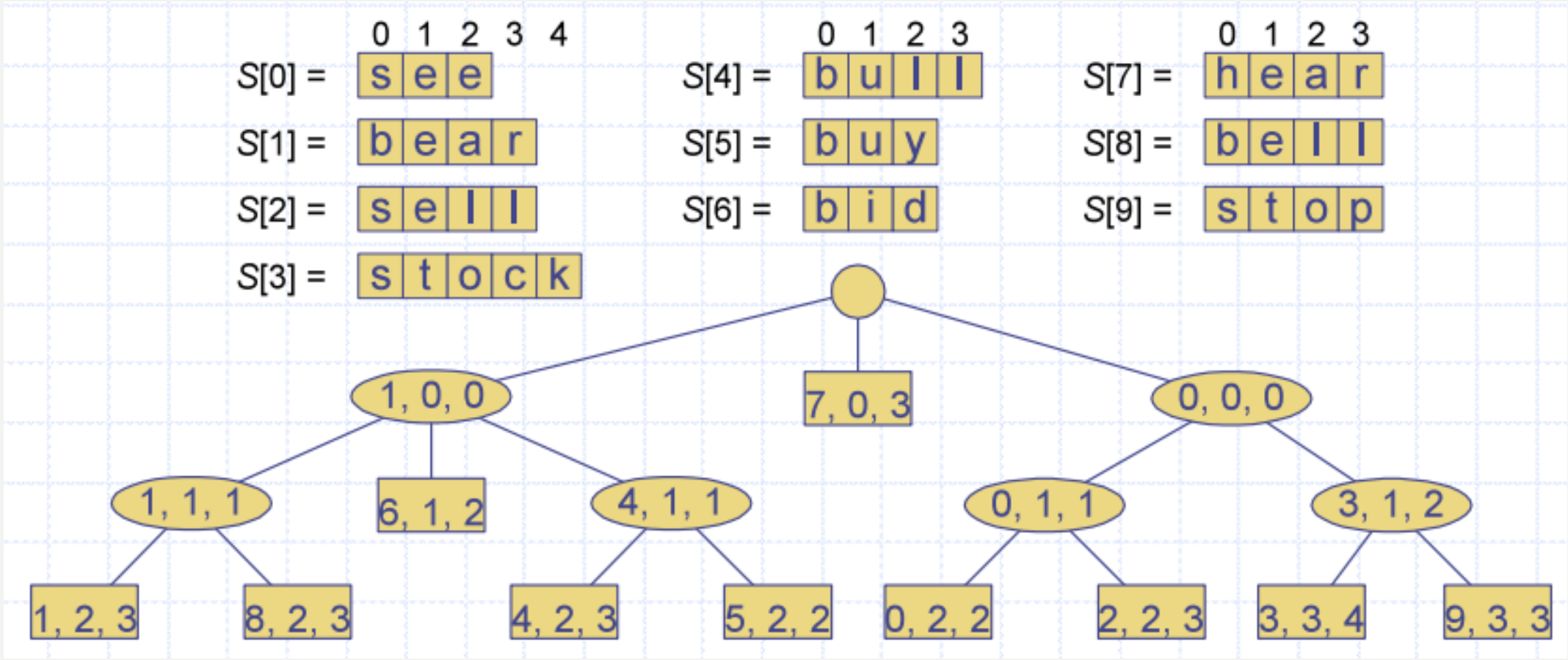
- It is obtained from standard trie by compressing chains of "redundant" nodes

Standard trie



Compressed trie

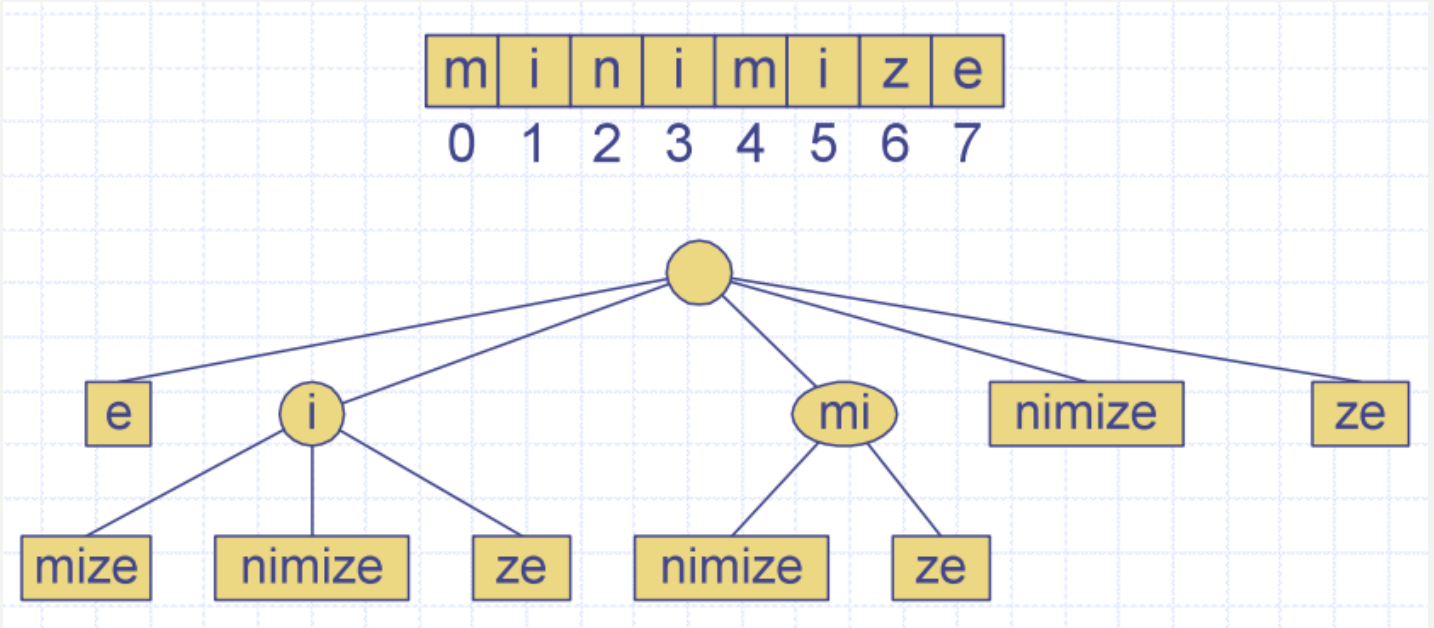
Representation of a Compressed Trie:



Suffix Tries:

- The suffix trie of a string X is the compressed trie of all the suffixes of X.
- A suffix tree **T** is a natural improvement over trie used in pattern matching problem, the one defined over a set of substrings of a string **s**.

Consider an example: The word: minimize



E
Ze
Ize
Mize
Imize
Nimize
Inimize
Minimize