

# \* Trie \*

Also called as keyword Tree. They are based on the prefix of a string.

Name Comes from - Retrieval.

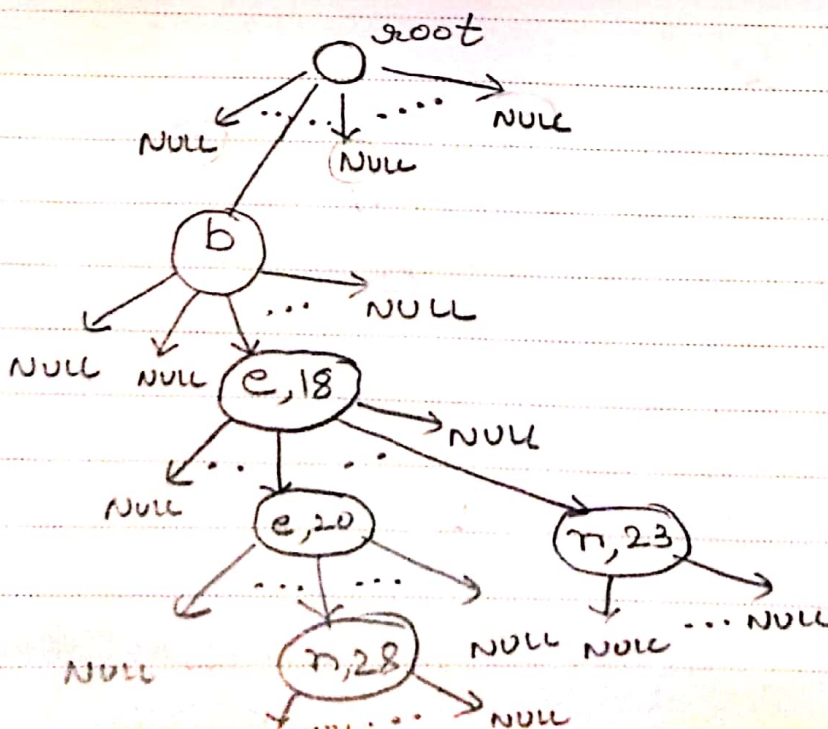
"The size of a trie is directly correlated to the size of all possible value that the trie could represent"

A trie is a tree-like data structure where in the nodes of the tree store the alphabets & strings can be retrieved by traversing down a branch path of the tree.

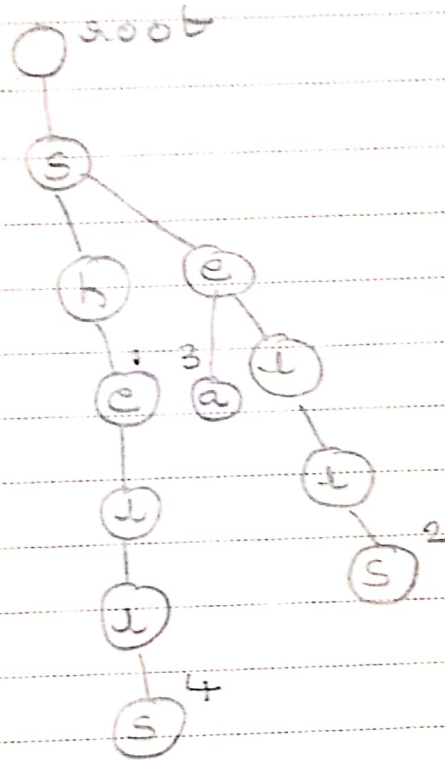
## Trie Creation :

keys:

be, 18  
bee, 20  
been, 28  
ben, 23

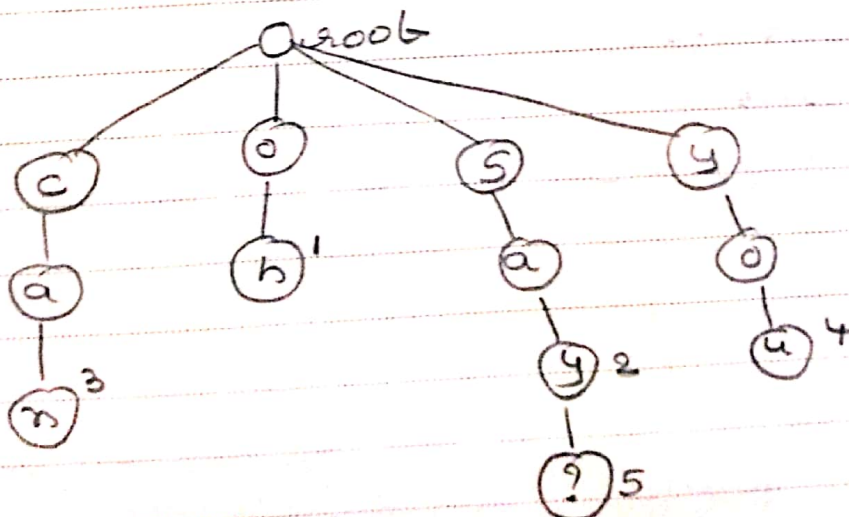


1. Create trie for:  
 she sells sea shells



Not all the  
 pointers are  
 shown in the  
 tree.

2. Create a trie for:  
 oh say can you say?



3. Create a trie for following and perform necessary operations as given.

Insert:

how, 1

much, 2

wood, 3

would, 4

a, 5

woodchuck, 6

chuck, 7

if, 8

could, 9

do, 10

can, 11

car, 12

chuckie, 13

howdie, 14

howdy, 15

Search for:

do → returns 10

how → returns 1

car → not found

Delete:

a → remove

ho → not found

wood → remove value 3

could → remove 'ould'





Scanned by CamScanner

## Advantages:

- Well suited data structure for predictive text, autocomplete dictionary, Spell checking etc.
- Pointers can be array of fixed indices
- Though they are similar to hash table, they don't need a hash function. Every branch path to a string's value will be unique to that key.
- As the tree, trie, grows in time, with size, there is less work each time to add a value.
- Adding intermediate nodes becomes a lot easier.
- The number of pointers in each case is fixed - constant value.
- If there are many common prefixes, the space is shared
- An immutable trie is compressible.
- Trie supports search, insert & delete operations in  $O(m)$  time where  $m$  is length of the key.

## Disadvantages:

- For a cambodian alphabet with 74 characters, the root node would contain 74 links to 74 other children & so on.
- Takes up a lot of space with null pointers.
- Longer words will grow the tree & some spaces may never get used.
- Larger Fanout when prefixes aren't shared.
- Implementation that has to manage each nodes pointers & values. A naive version uses a hashmap to track and manage its children.