


↳ Type ahead / Autocomplete →

↳ Netflix / Amazon

Harry Potter

→ Common feature

↳ Inside your mobile phones

↳ Sa

→ Sam

→ Samneet

→ Samket

→ Sahil

⋮

HP	→	Priscu	→
HP	→	Gob	→
HP	→	DH	→
⋮			

}

Acc. to students

↳ String Matching

↳ Hashing

↳ New Data Structure - TRIE

(1) String Matching \rightarrow Given a text and a pattern

find the occurrence of pattern in text.

text \rightarrow abcababccab
pattern \rightarrow cab

$\left. \begin{array}{l} \rightarrow \text{Rabin Karp} \\ \rightarrow \text{KMP} \\ \rightarrow \text{Zalgo} \\ \rightarrow \text{aho corasick} \end{array} \right\} \rightarrow O(t + p)$

\hookrightarrow Let's say in Data store when we store all the

movies of amareen prime. & we have total N movies

& max length of name of any movie is l .

$\hookrightarrow O(N \times (l + l))$ \leftarrow time for one search

(11) Use Hashing \rightarrow Use hash tables $\langle \text{key}, \text{value} \rangle$

uses the mechanism of Hashing

\hookrightarrow Generally for storing or search a string in hash table, we require $O(\text{length of string})$

If you have N string in HT, to search any or string you require $O(\text{length of string}) \rightarrow$ for one query

2 of each character takes 1 unit memory

Let's say the max length string in your database is 'L', and min length is 1

Consider, you store every thing in lower case

→ How many unique one length strings we can have? (26)

(a, b, c, ..., i, ..., z) → 26 unit

→ How many 2 length strings are there? 26 ¹ 26 ² → 26^2

→ space → 2×26^2

1 → 2 unit

$26^2 \rightarrow 2 \times 26^2$

→ 1 length string space → 1×26^1

2 length string space → 2×26^2

3 length string space → 3×26^3

$\begin{array}{r} \textcircled{3} \\ \hline 26 \quad 26 \quad 26 \\ \hline 3 \times 26^3 \end{array}$

l length string space → $l \times 26^l$

total space → $1 \times 26^1 + 2 \times 26^2 + 3 \times 26^3 + \dots + l \times 26^l$

In asymptotic analysis → Space complexity → $O(l \times 26^l)$

final solution

$\emptyset \rightarrow \text{null}$

TRIE

prefix tree

- add
- Sam
 - Sameer
 - Sanket
 - Sahil
 - Saharsh
 - Samyak
 - mohit
 - mohan

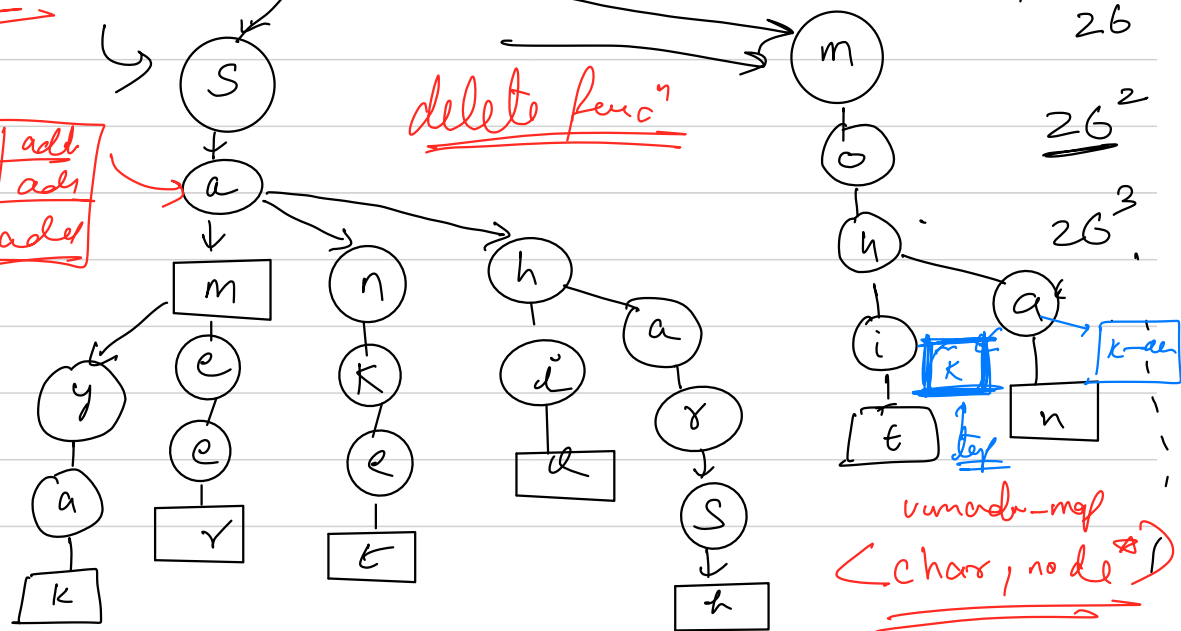
query \rightarrow sq

Samy

apple
ake

root \rightarrow uppermost node with no parent

m	add
n	add
h	add



delete func

terminal

umabr-map
chars, node

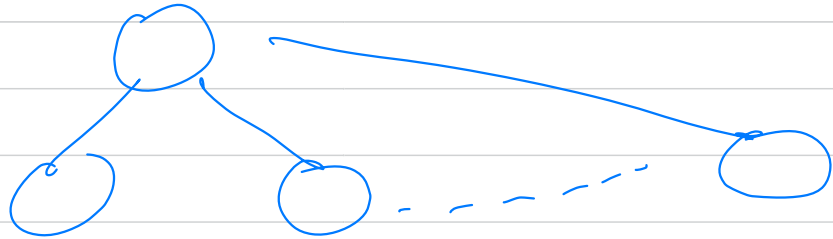
→ Trie is a prefix tree

→ That means it is a hierarchy based data structure.

→ This is an n -ary tree

generic
tree

one parent can have n child



Trie is space optimised

$$\text{Space} \rightarrow 2G + 2G^2 + 2G^3 + \dots + 2G^d$$

$$\underline{\underline{O(2G^2)}} \longrightarrow$$

Tr:

URL

→ <https://www.google.com>

<https://www.netflix.com>

<https://www.unacademy.com>

