→ Why a trie?:

How to build a trie

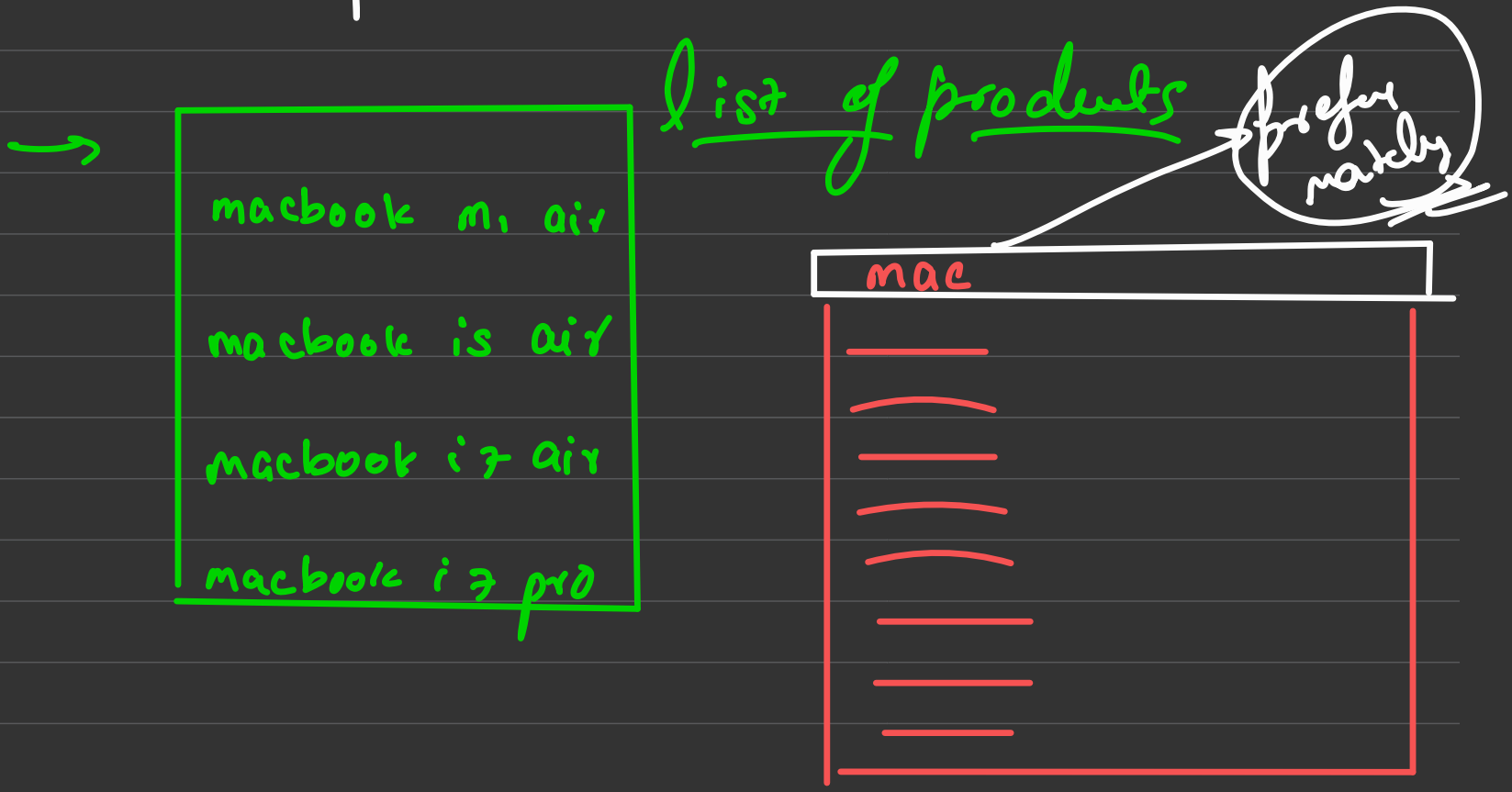
basic application of trie

is a Binary Trie

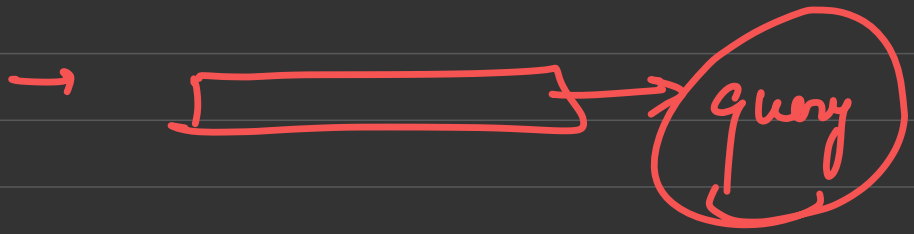
Persistent trie

is

→ Autocomplete ←



Let's see a simpler problem



Check whether it is present in the storage or not -

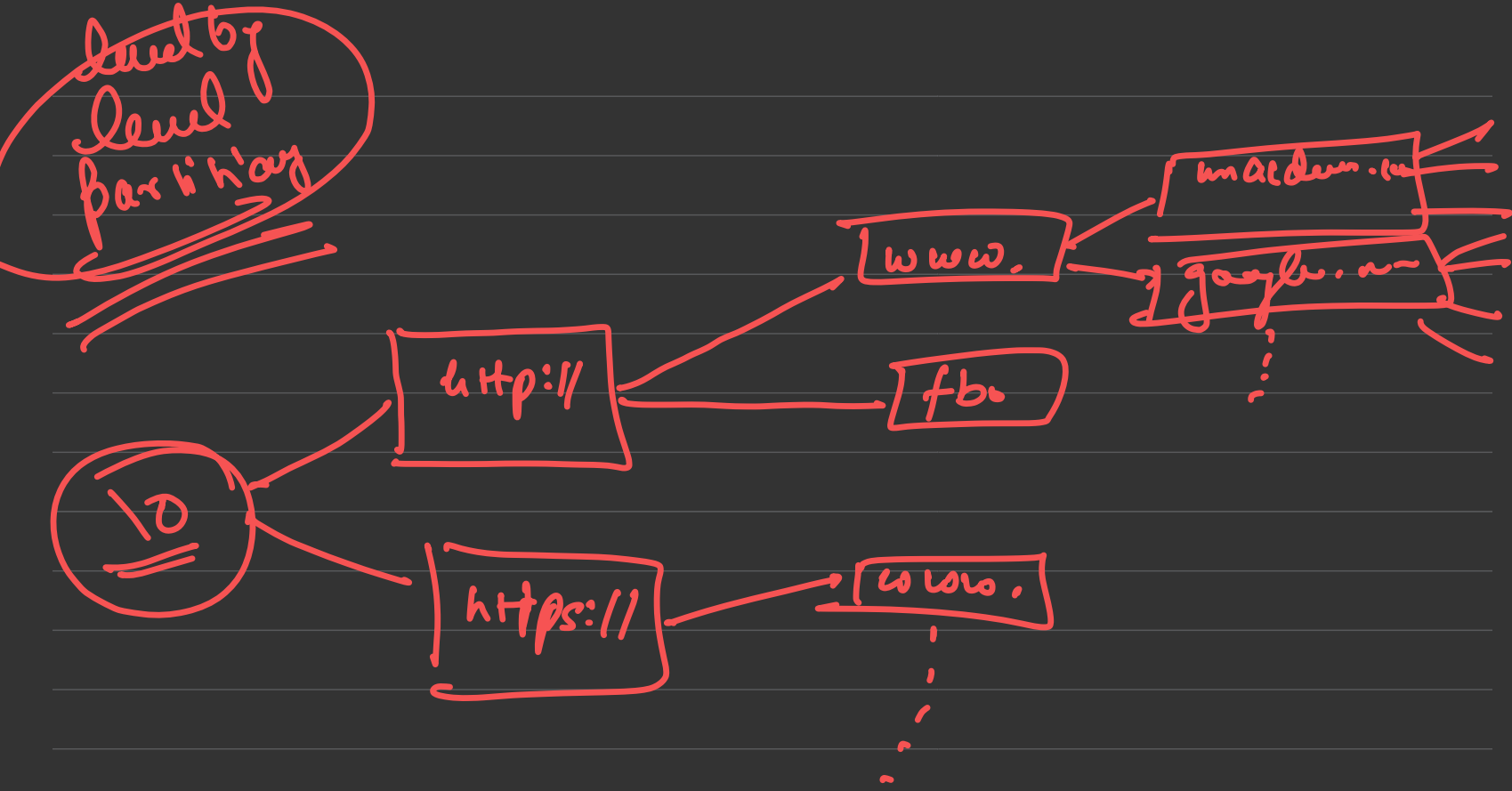
URLs ← autocomplete of URL

http://www.inacademy.com/classroom....

https://www.

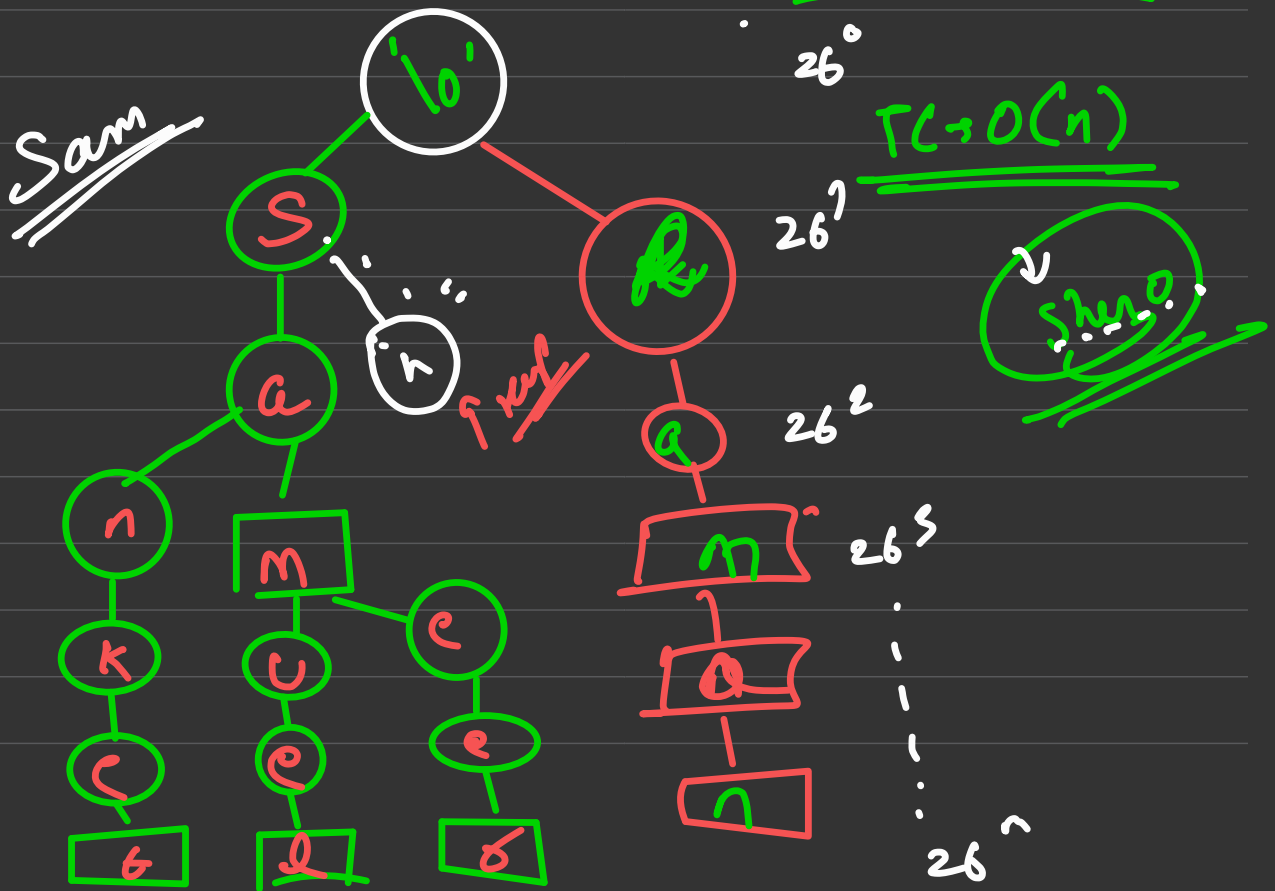
ftp://

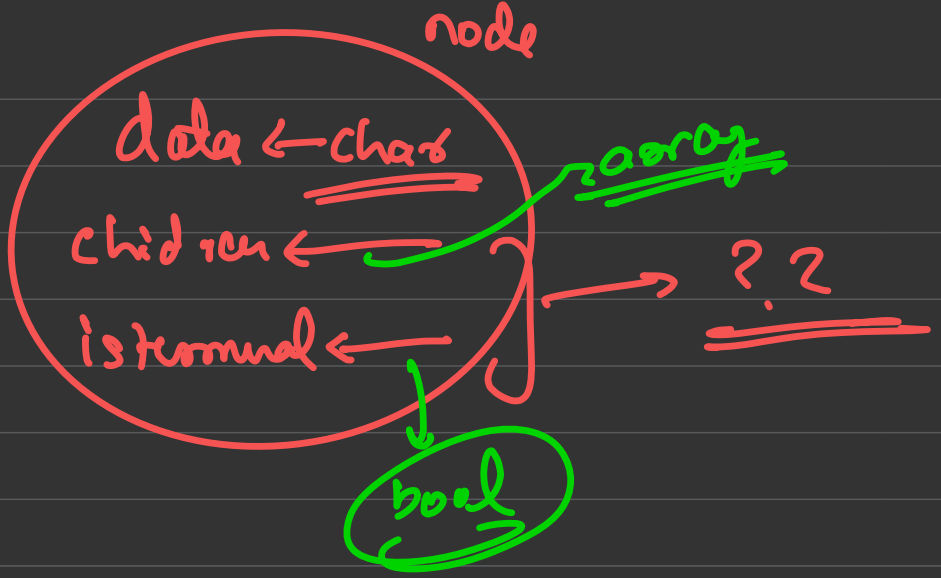
http://hardum.


Level by
level
parsing
www.
http://
fb.
IO
https://

n-ary tree ← TRIE ← prefix tree

Sam
Sanket
Samuel
Sameer
Samantha
Shero
Raman
Kama
Kam




node
data ← char
children ← array
isTerminal ← bool
??

Hashmap

→ We will store n strings of max m length.

Assume 1 char = 1 Byte → only small alphabet

→ We will have variety in string length from 1 - m

→ How many unique 1 length string are stored in HM.

↳ space → 26 bytes ↑ 26

Space → 2 length → $\frac{26}{\uparrow}\ \frac{26}{\uparrow}$ → 2×26^2

$1 \times 26^{1} + 2 \times 26^{2} + 3 \times 26^{3} \cdots\cdots m\,26^{m}$

$26 + 26^{2} + 26^{3} - - - - - - - 26^{n}$

How to build a Trie