The (Trie) Data Structure :→

* Telephone Directory
* Word Dictionary
* Longest Common Prefix

Aarya
Aaryan

Insert → name → Dictionary

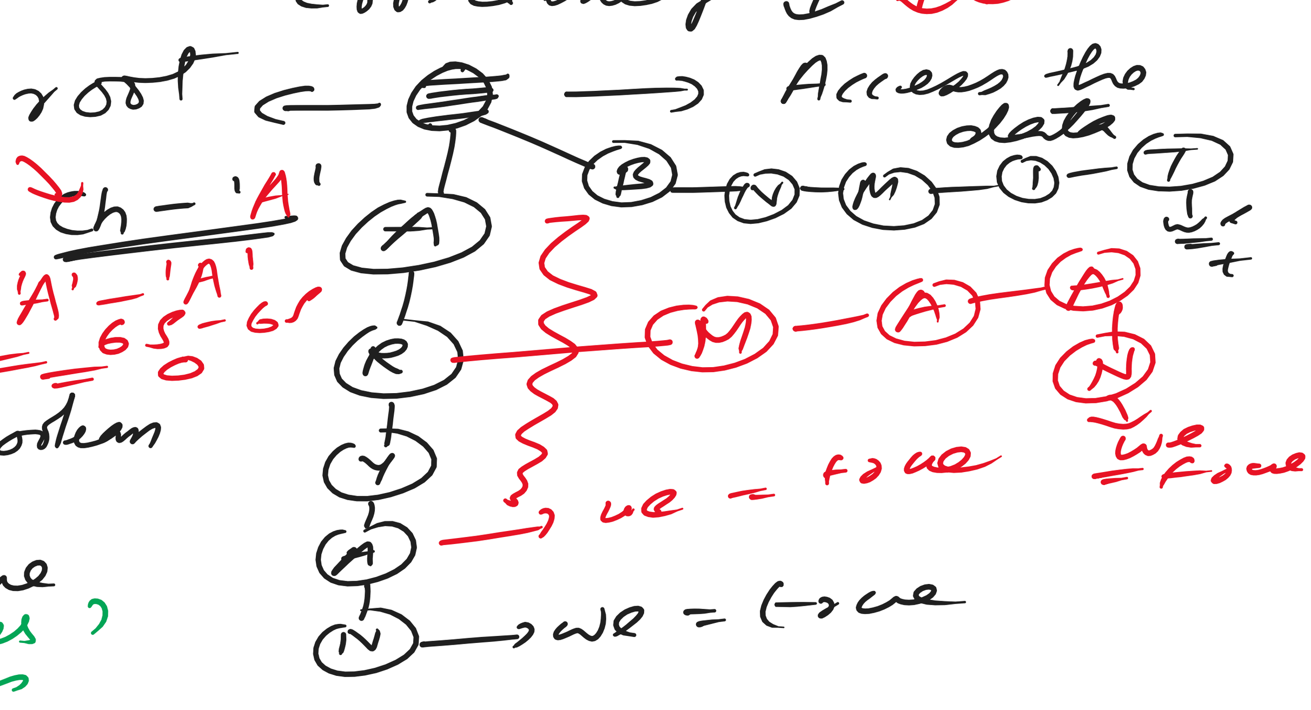
HashMap<String, Boolean>

A, t → ("Yathin", true);
A, t → ("Kamal", false);

1000 words → 2000 spaces
Efficiency ↓ ↓↓

ch

(i) ARYAN
(ii) ARYA
(iii) ARMAAN
(iv) BNMIT

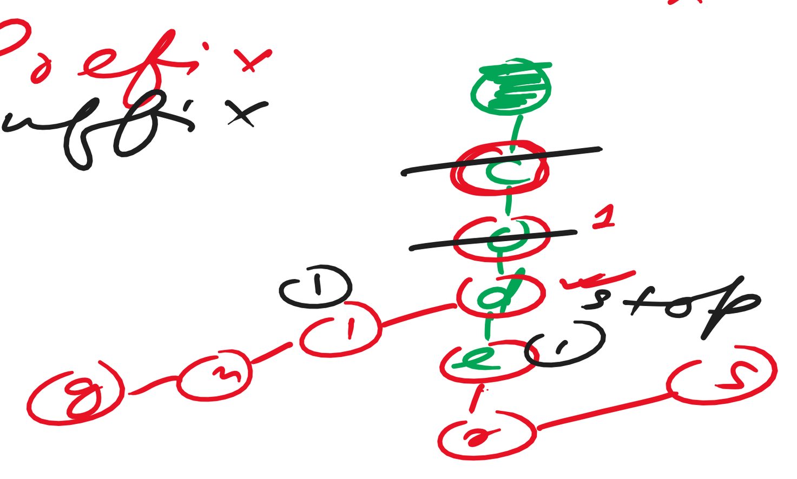


wordEnd → boolean
false
↳ true
0-25
A (26 nodes)
52 nodes

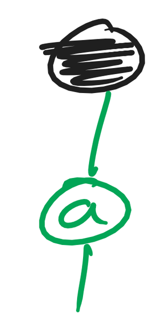
(i) insert (ii) search (iii) delete ×

Longest Common Prefix
Suffix ×

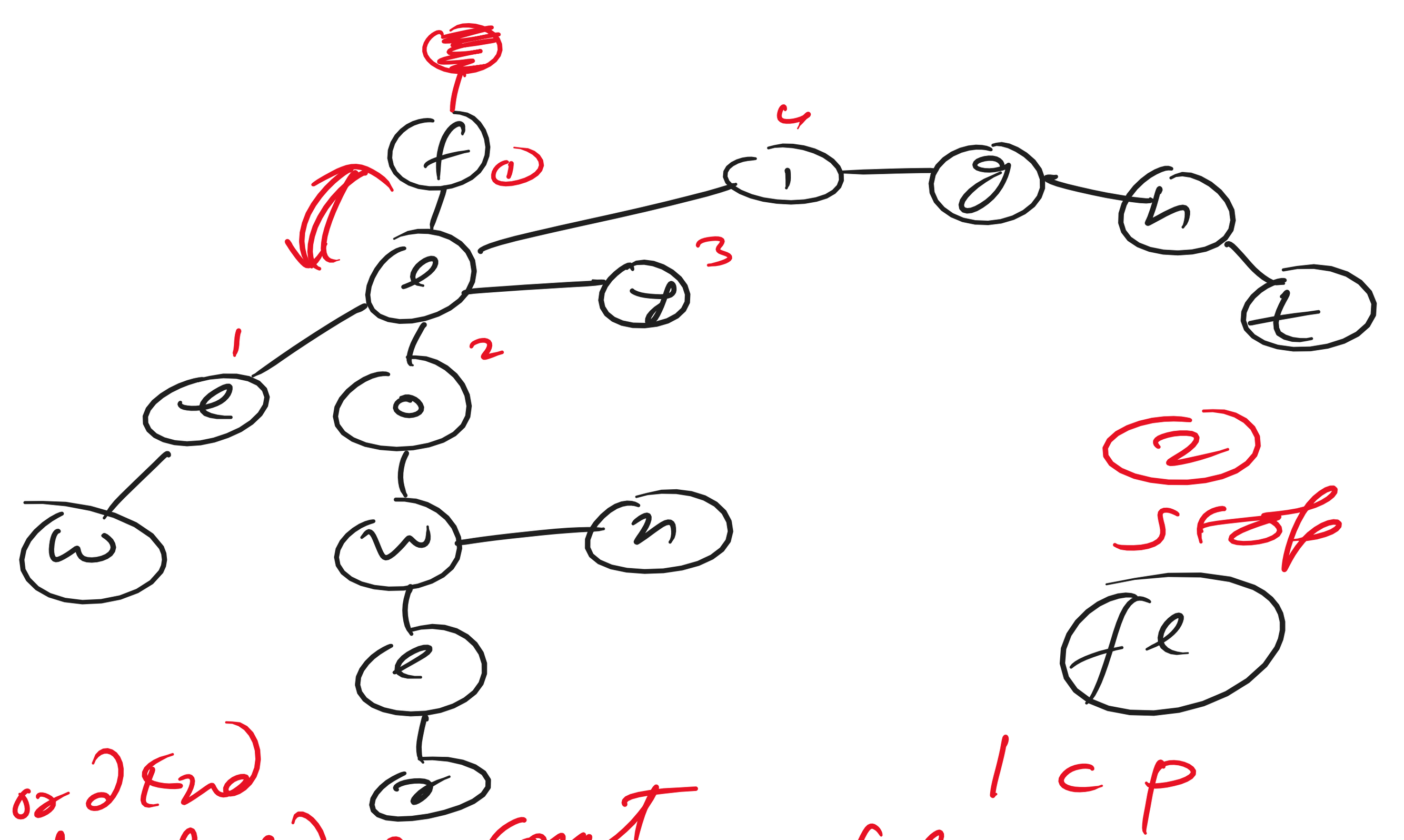
Code
Coder
Coderer
Coding



curr
↓
curr



a, r, y, a, n
'b − 'a' = null
= '
0 —— 25
[not present]
b ——→



wordEnd
int childrenCount

(2) stop
fe
lcp
fe

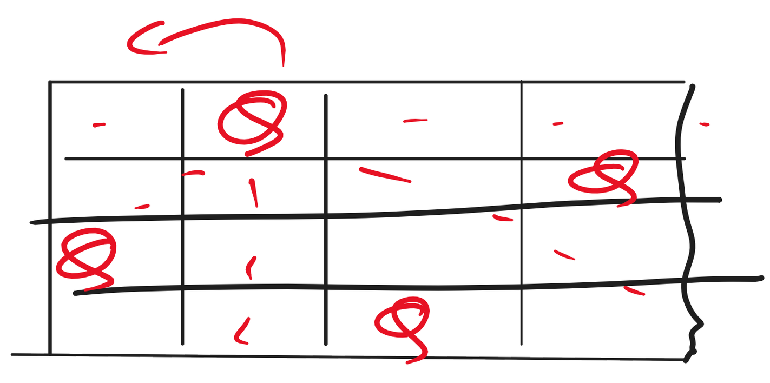
Back Tracking :→ Application of Recursion

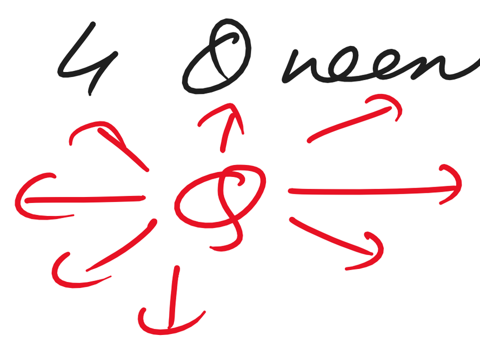
* N Queens
* Sudoku Solver
* Rat in a Maze
* Subsets of a String
* Subsets of an array → Power Set

a, b, c
{ }
×

{1, 2, 3}
{1}, {2}, {3}, {1, 2}, {1, 3},
{1, 2, 3}, {2, 3}, { }
$2^n \to 2^3 = 8$

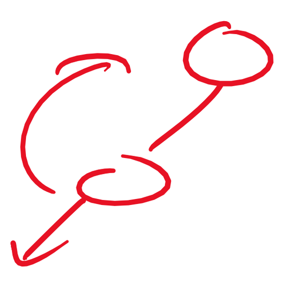
4x4 Matrix → 4 Queens





attack ×
4x4

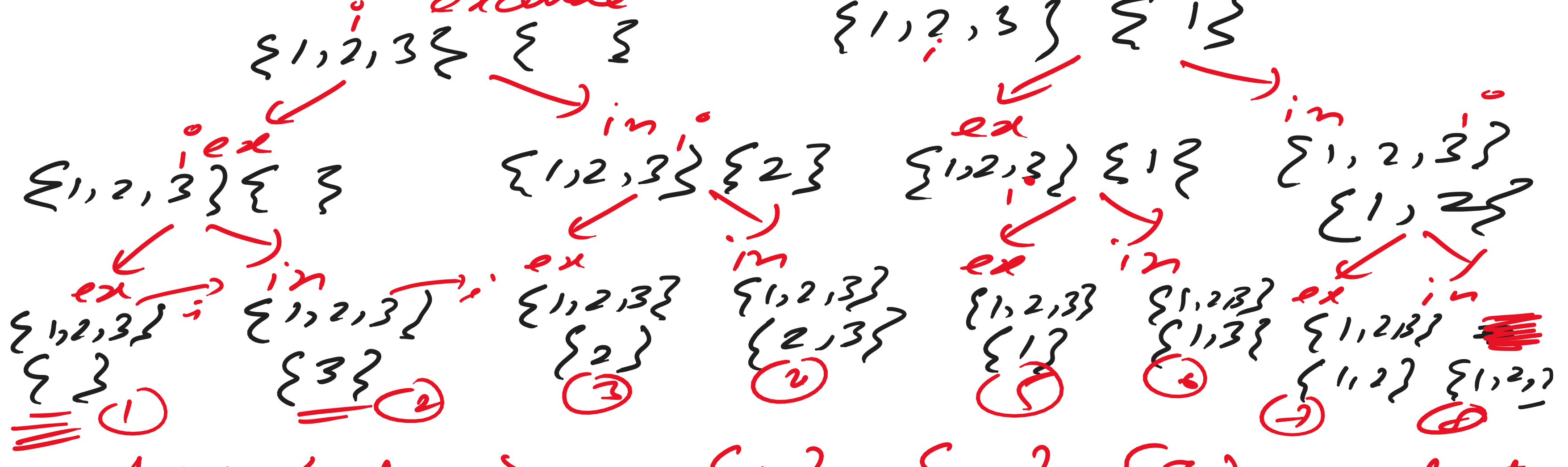
Recursion



Leet Code (78) → Set of all subsets of an array: ⇒ Power Set

(n = 3)
$\{1, 2, 3\} - 2^3 = 8$ subsets

Recursion Tree {1, 2, 3} → i/p { } → o/p



List<Integer> = {1}, {2}, {3} → output
List<List<Integer>>