Trie Data Structure :→

* Phone Contact List / Phone book
* Word Dictionary (a-z)
* Web Browser Autocomplete Feature

Why? {word}

Map <String, Boolean>

(Anup, true); → v1

(Ajay, false); → v2

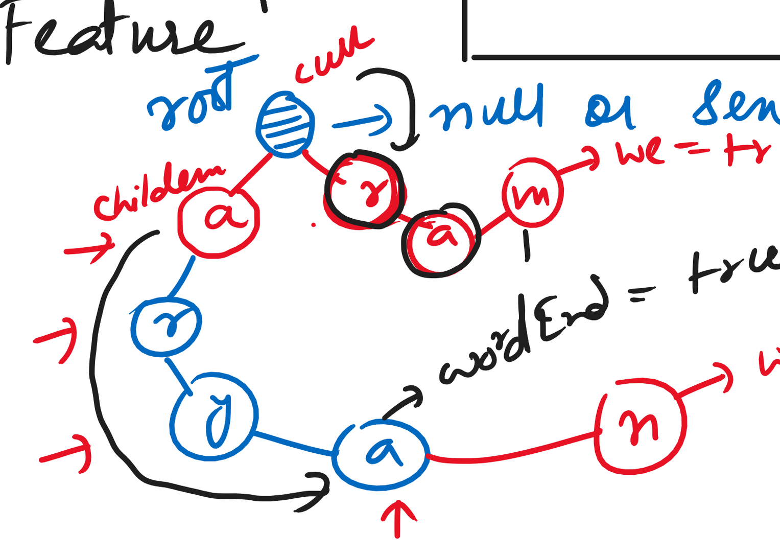
Memory Efficient *

ch

arya → string
aryan
aryaman
ajay

wordend = false

ram



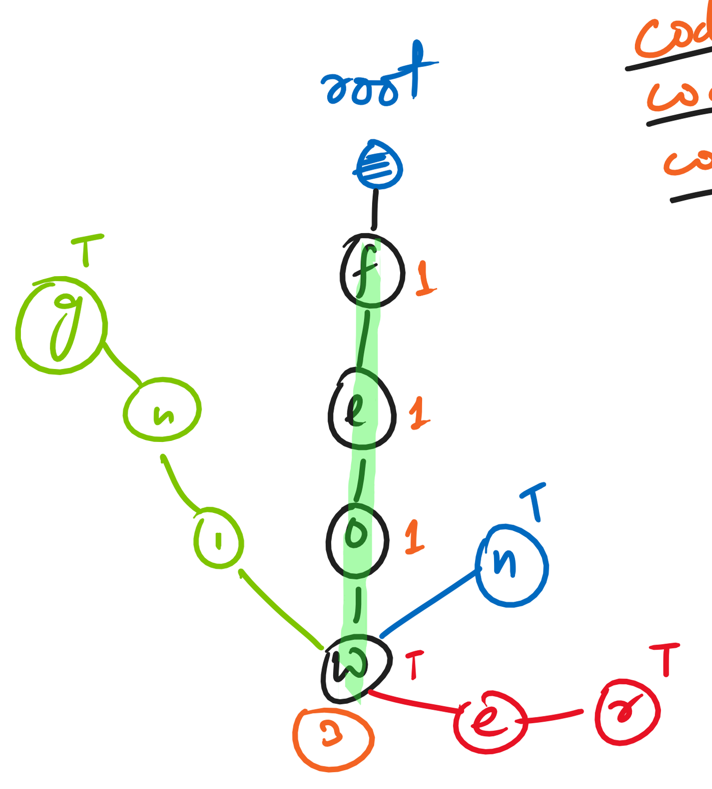
insert ✓
search ✓
delete ×

LCP → Longest Common Prefix :

flow
flower
flowing
flown

lcp = flow

word end = True
childen count



coder
coders
coding

flow
flown
flower
fly

(childencount == 1)
>1 stop

Introduction to Greedy Algorithms :→

Most frequently asked questions in Coding Interviews:

1. Minimum number of coins.
2. Minimum cost of connecting ropes.
3. Minimum absolute difference or (Chocolate Distribution Problem)
4. Huffman Encoding
5. Fractional Knapsack
6. 0/1 Knapsack
7. Activity Selection Problem
8. Job Scheduling Problem
9. Minimum number of Platforms
10. Gas Station Problem
11. Lemonade Change Problem
12. Minimum Arrows to burst Balloons
13. Nikunj & Donuts
14. Sliding Window (Max/Min)

Minimum number of Coins to reach a value :

Sort coins = {1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000} ← last

V = 91 >= Coins.get(i)

```
for( i = coins.size() - 1; i >= 0; i-- ) {
    while (V >= coins.get(i))
        V -= coins.get(i)
}
```

array = (50, 20, 20, 1) 4 Coins

100 max no coins 1×100

91 − 2000
− 1000
500
100
91 − 50 = 41
41 − 20 = 21
21 − 20 = 1
1 − 1 = 0

V = 31 = 20, 10, 1

* Minimum number of Platforms :→

arrival (Sort) = { → 900, 940, 950, 1100, 1500, 1800 };

departure = { 910, 1200, 1120, 1130, 1900, 2000 };

Sort → 910, 1120, 1130, 1200, 1900, 2000

platform
++ 1
-- 0
++ 1
++ 2
++ 3
-- 2
-- 1
-- 0
++ 1
++ 2

-- 1
-- 0

max platform 1 ⇒ ③

* Activity Selection Problem :→

Entity <var1 (start), var2 (finish)>
→ Comparator *

* Given a list of activities with their start & finish times, calculate the maximum number of activities that can be performed by a single person at a given time if he/she can perform only one activity at a particular time.

Activity	Start	Finish	Sort (Finish Time)	Output
A1	5	7	A3 (1, 4) ✓ 0	ls A3 (1, 4)
A2	8	9	A6 (3, 5) i	A1 (5, 7)
A3	1	4	A5 (0, 6) i	A2 (8, 9)
A4	5	9	A1 (5, 7) i ls	
A5	0	6	A5 (5, 9) i	
A6	3	5	A2 (8, 9)	