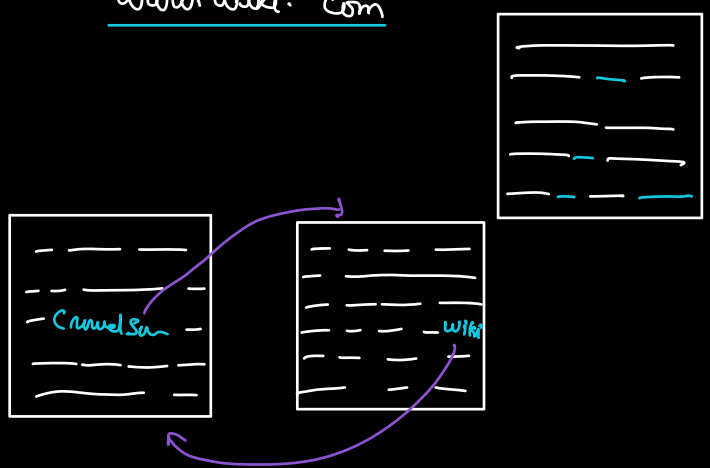
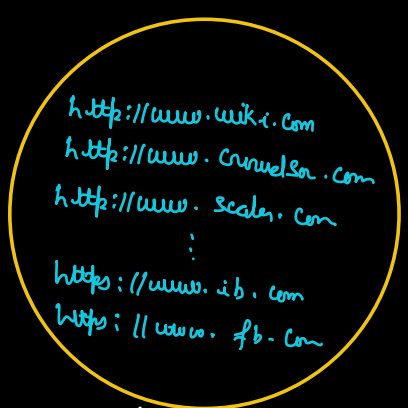
Facebook

www.wiki.com



Set



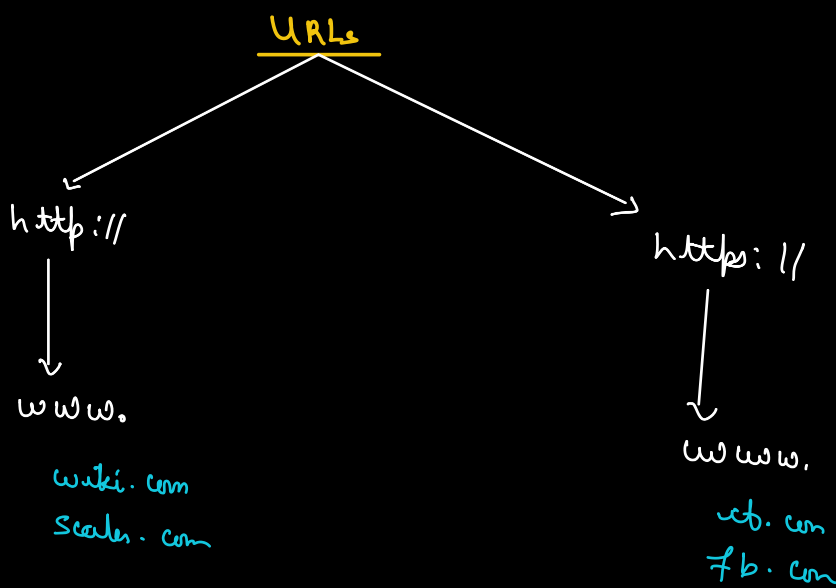
http://

www.wiki.com

www.scales.com

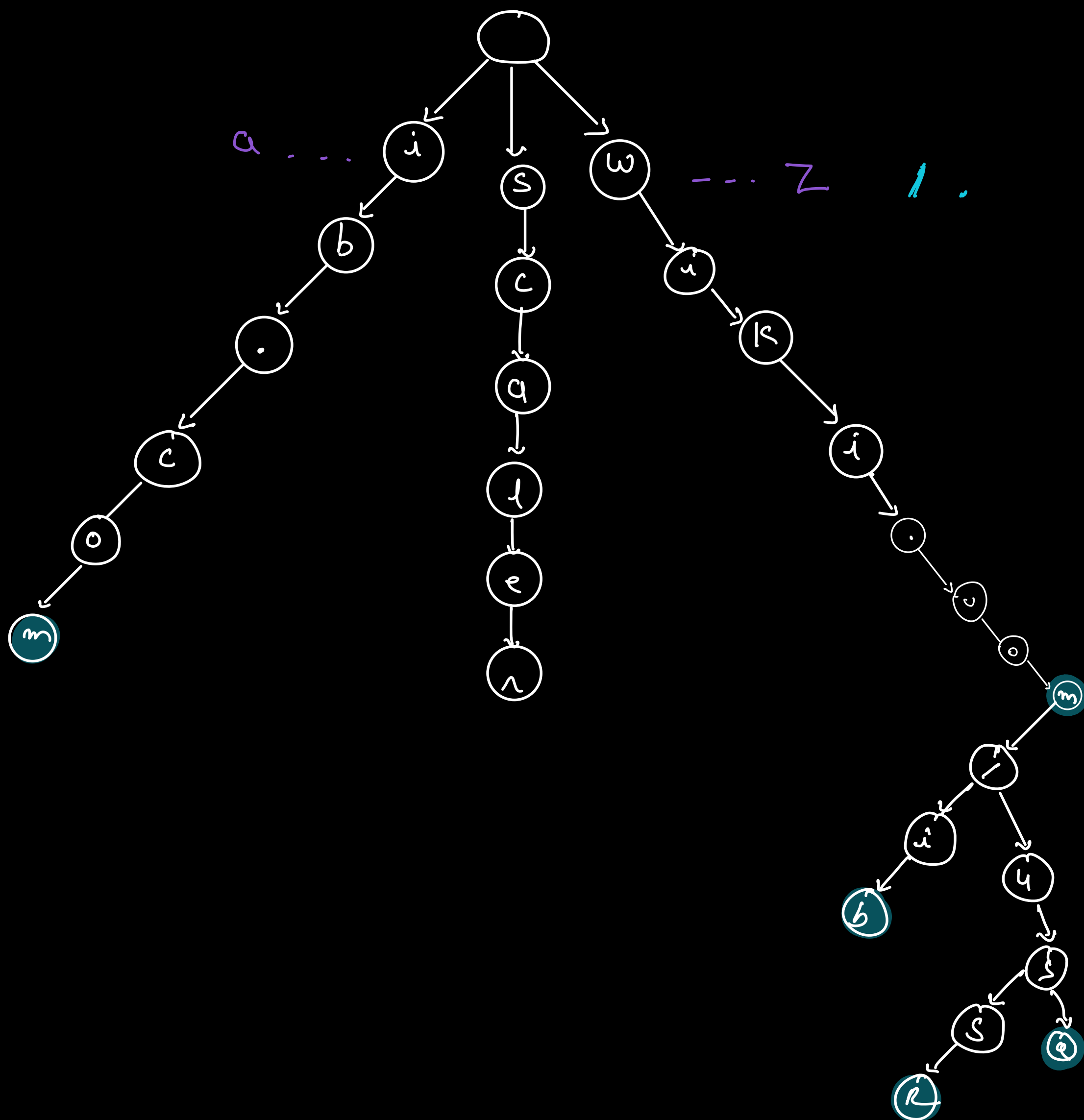
https://

www.ib.com



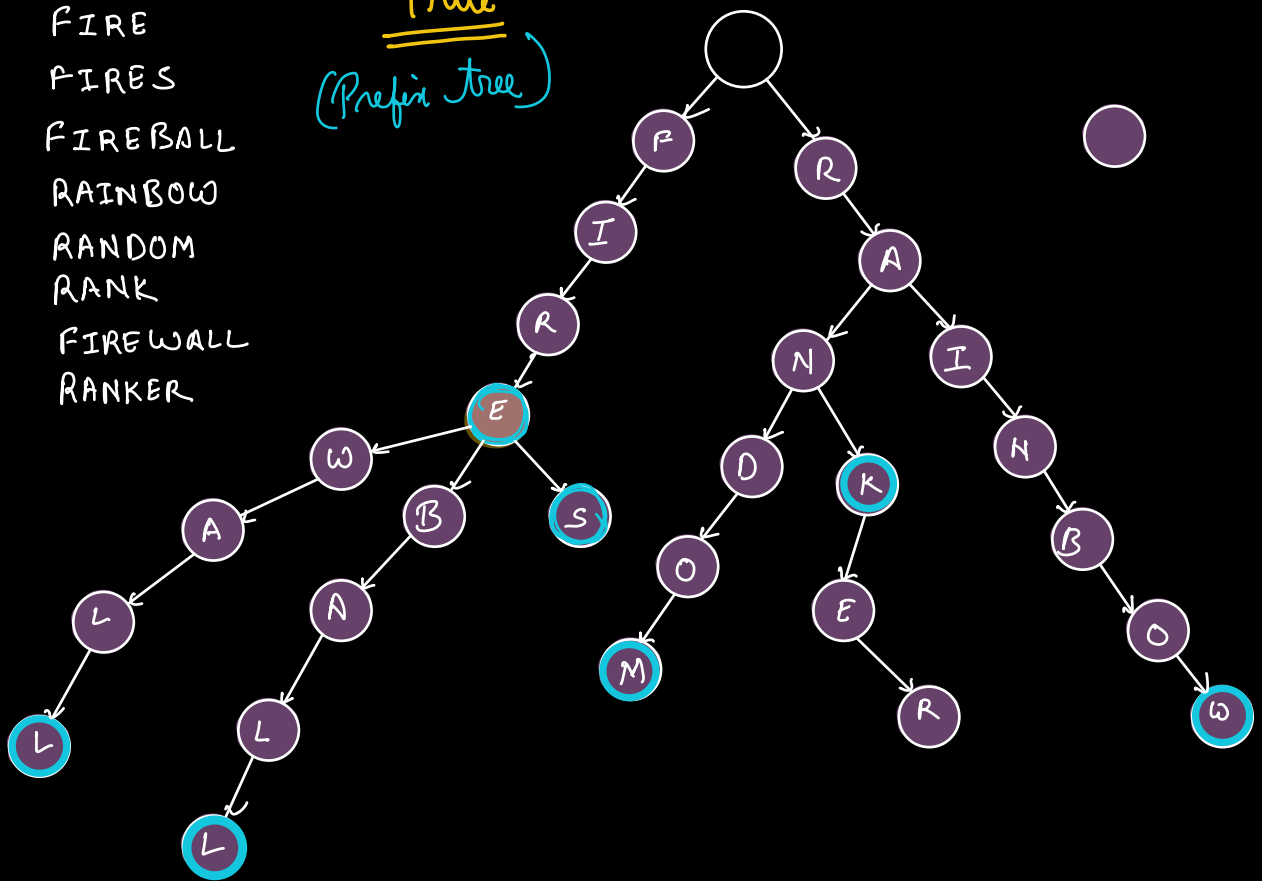
wiki.com
wiki.com/india
wiki.com/usa
wiki.com/music
wiki.com/sachin
⋮

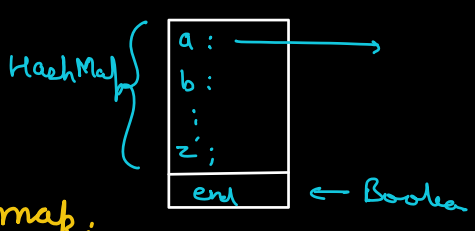
interviewbit.com
interviewcake.com
interviewace.com


a . . .
. . . z

FIRE
FIRES
FIREBALL
RAINBOW
RANDOM
RANK
FIREWALL
RANKER

Trie
(Prefix tree)






```
Class TrieNode {

    HashMap<Character, TrieNode> map;

    boolean end;

    Public TrieNode() {
        map = new HashMap<Char, TrieNode>();
        end = false;
    }
}
```

```
Class Trie {
    TrieNode root;
}
```

abc

abde

abcd

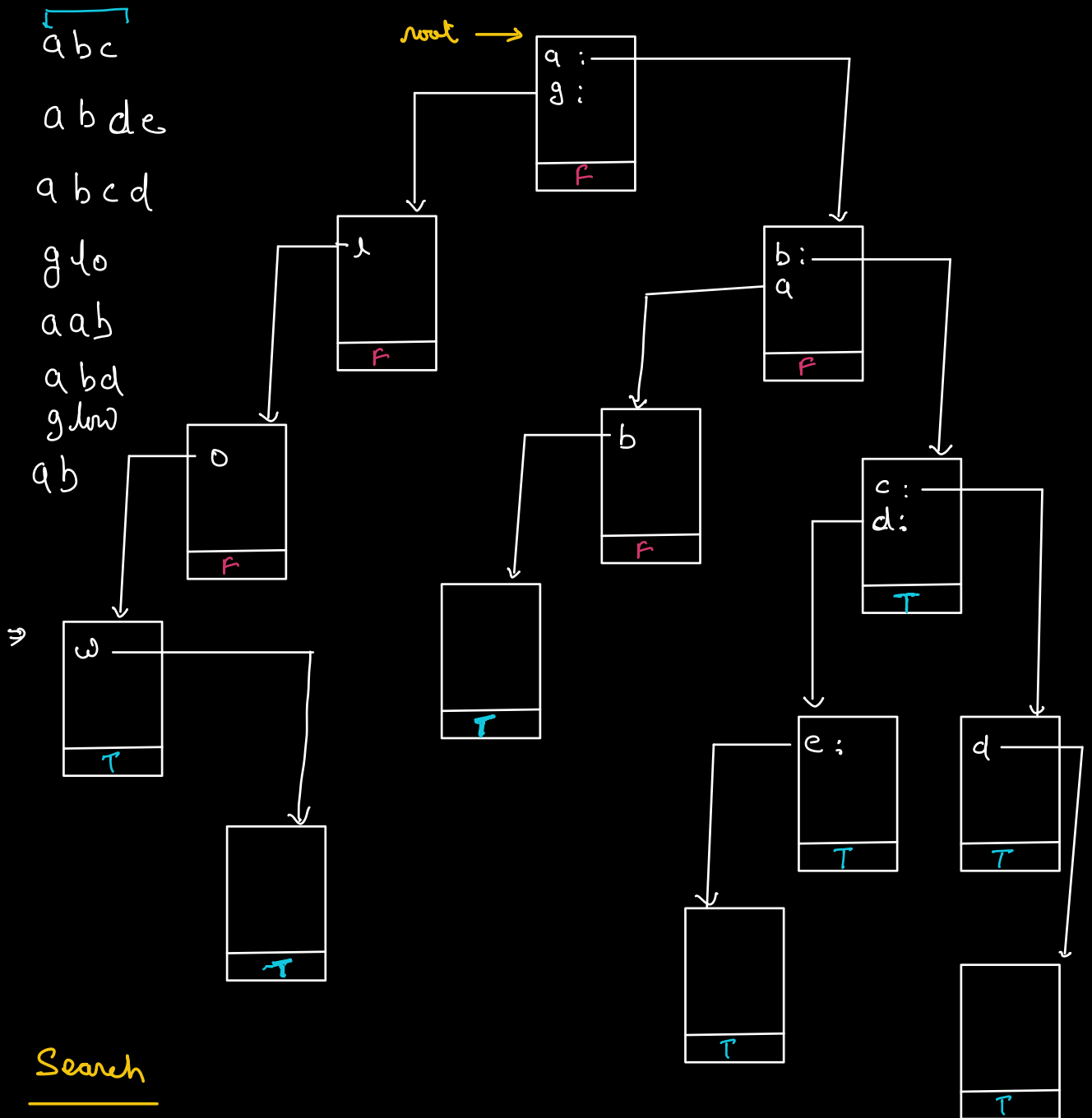
glo

aab

abd

glow

ab



Search

abcd ✓

abde ✓

abe ✗

gl ✗

```
void insert (word) {
    TrieNode curr = root;
    for (i=0; i< word.length() ; i++) {
        ch = word.charAt(i);
        if (!curr.map.containsKey(ch)) {
            curr.map.put(ch, new TrieNode());
        }
        curr = curr.map.get(ch);
    }
    curr.end = true;
}
```

TC : O(l) — l → length of word

SC : O(l)

```
boolean search (word)
    TrieNode curr = root;
    for (i=0; i< word.length() ; i++) {
        ch = word.charAt(i);
        if (!curr.map.containsKey(ch)) {
            return false;
        }
        curr = curr.map.get(ch);
    }
    return curr.end;
}
```

TC : O(l) — l → length of word

Soln / BS

↓

Interview window

(Not trie) ×

Q Given an array of words. Return an array of strings containing the smallest unique prefix for every word.

[cat, dog, rat, tiger]

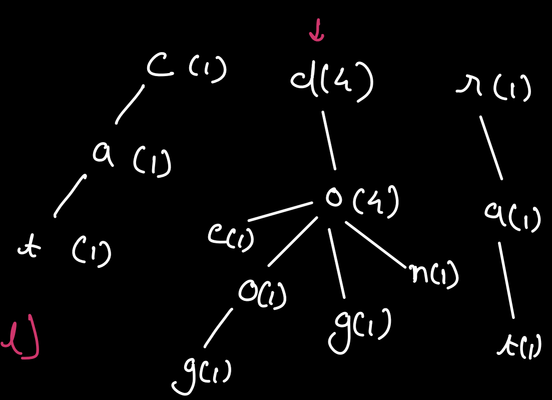
[c, d, r, t]

[cat, dog, rat, doog, doe, donkey] N

[c, dog, r, doo, doe, don]

~~[cat, cats]~~ ×

cat



TC : $O(ln) + O(ln) \Rightarrow O(nl)$

SC : $O(ln)$

Google Given an array. Return the pair with maximum XOR.

A: 25, 10, 2, 8, 5, 3

25^5 ⇒ 28

5 : 101

A: 1, 2, 3, 4, 5, 6, 7

3^4
6^1
2^5
⇒ 7

1 : 001
2 : 010
3 : 011
4 : 100
5 : 101
6 : 110
7 : 111

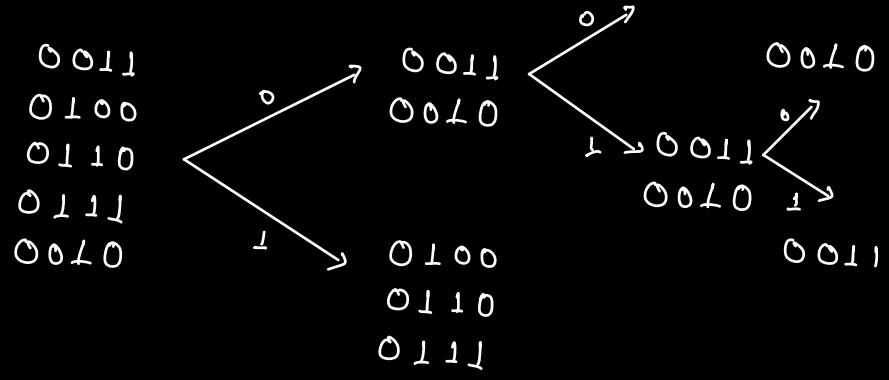
Brute Force — iterate over all possible pairs

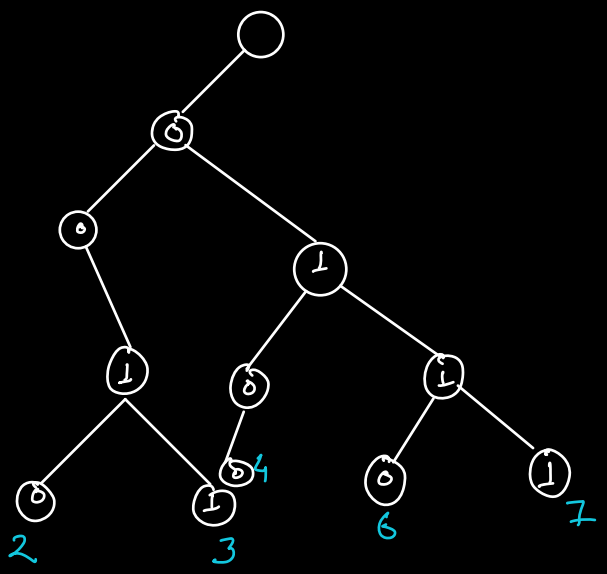
TC: $O(N^2)$

Given a no N, & an array.
Find the element from array that gives max XOR with N

N=5, [3, 4, 6, 7, 2]

0101
^ 0010
0111





$N = 6$ 0110

0011

Tc : $O(bN) \Rightarrow O(N \log M)$

No of bits in datatype

$\log_2 M$