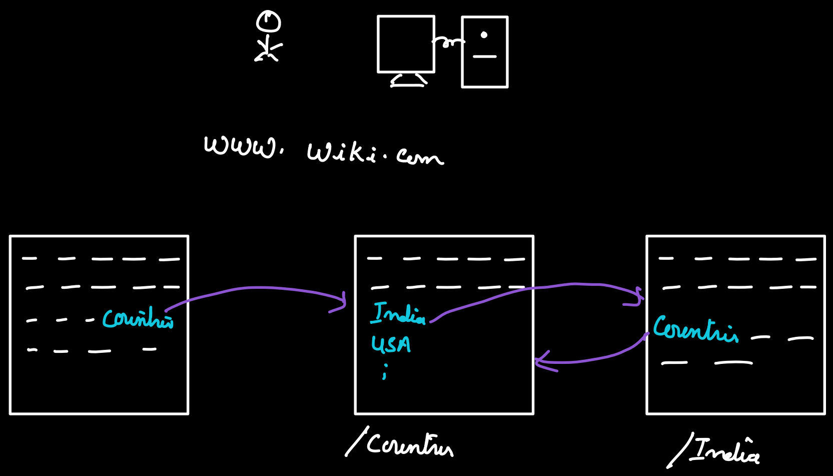
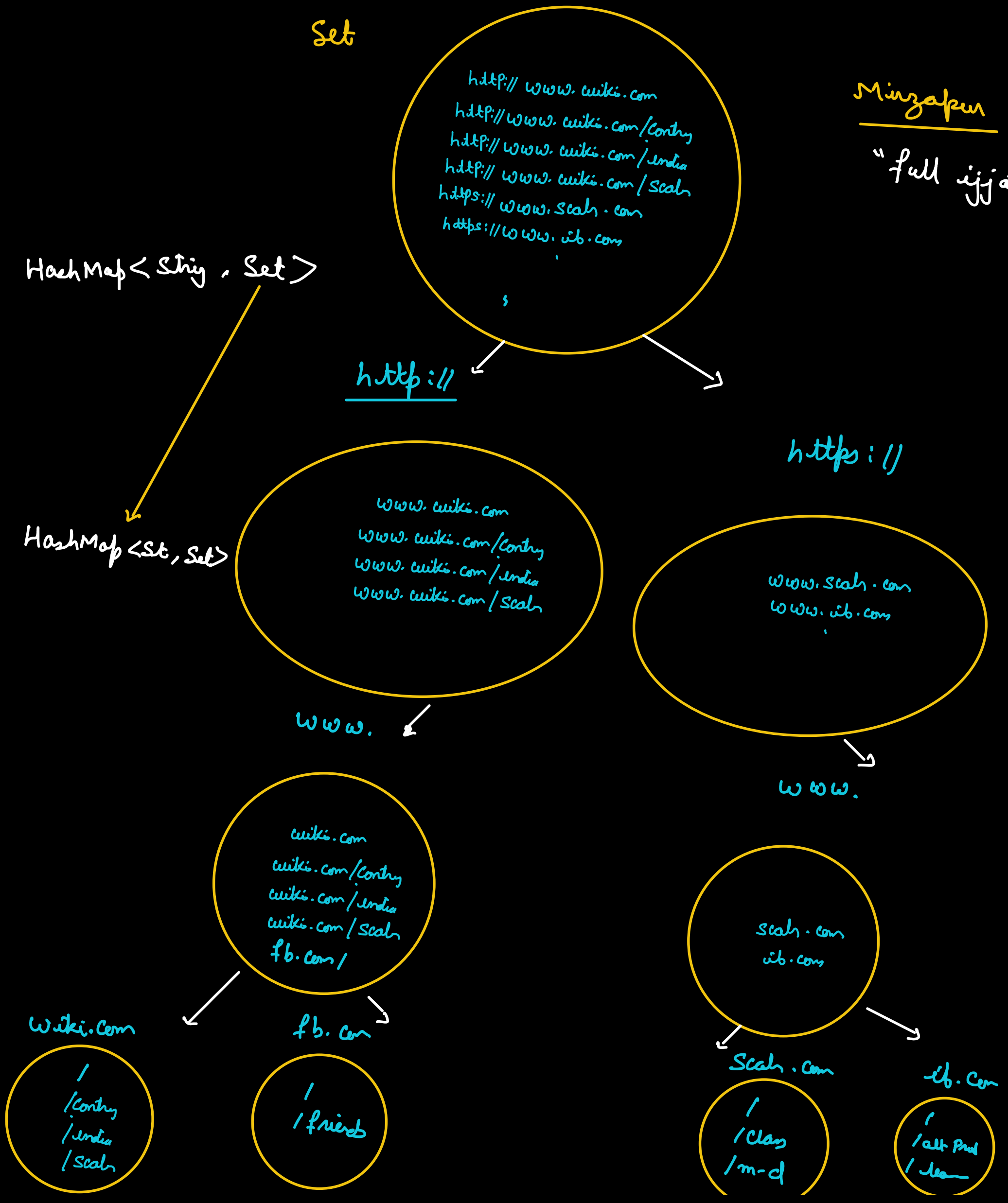
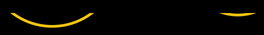
Facebook



Mirzapur

"full ijjat"





interview bit . com

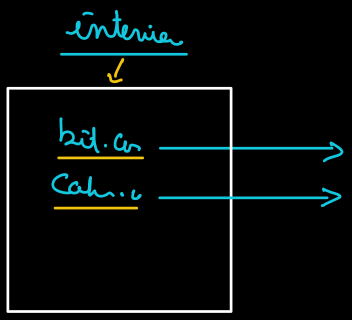
interview cake . com

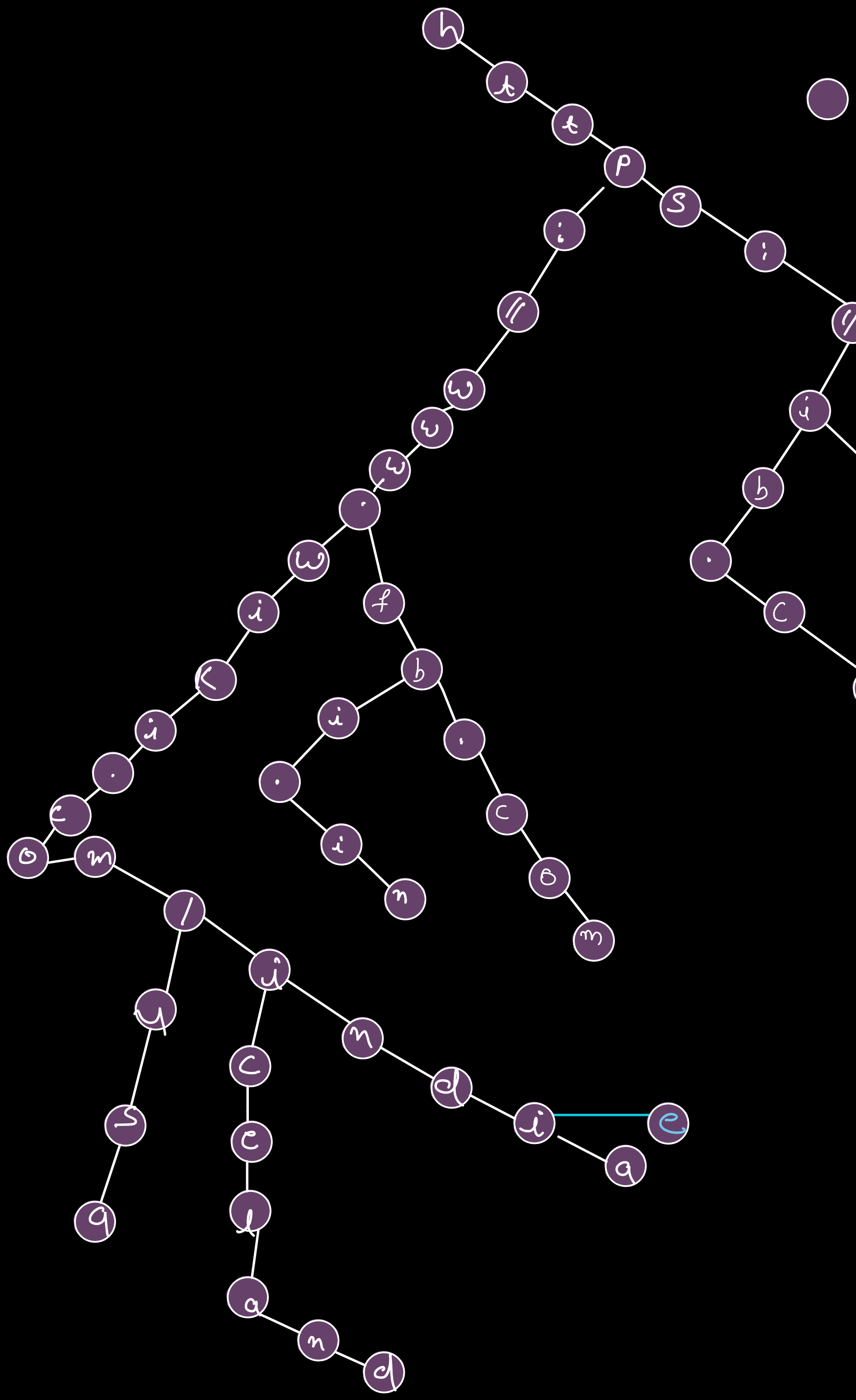
interview ace . com

interview delo . com

interview dedo . com

intervene . com

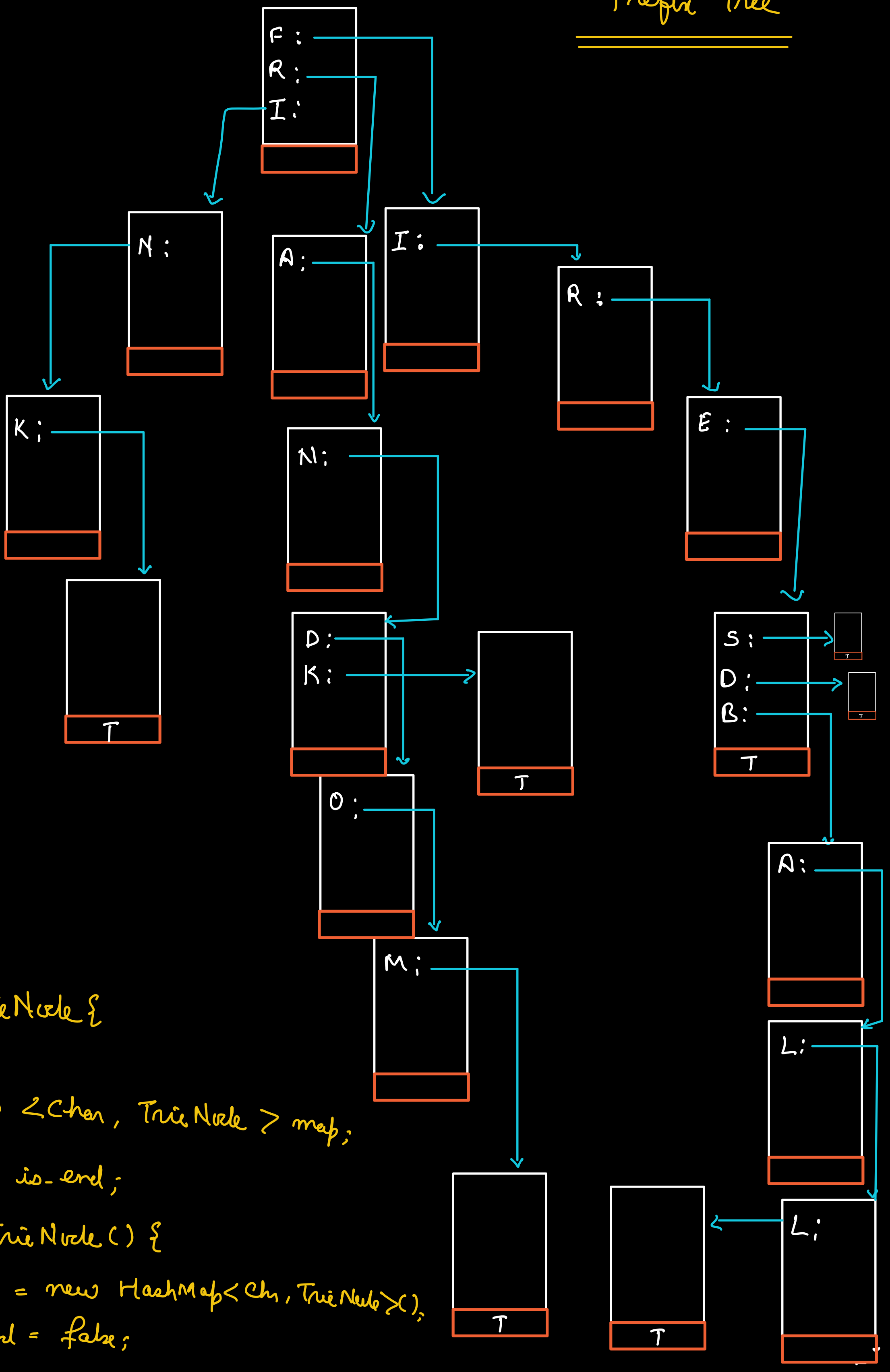



h t t p
p : // w w w . w i k i . c o m /
/ y s q
/ j c e l a n d
/ j n d i a
/ j n d i e
. f b . c o m
. f b i . i n
p s : / i b . c o m
p s : / i p l . i n

INK
FIRE
FIRES
FIRED
FIREBALL
RANDOM
RANK
FIREWALL
RANKER
DIRE

FIR

Prefix Tree



```
Class TrieNode {

    HashMap <Char , TrieNode > map;

    boolean is_end;

    Public TrieNode() {

        map = new HashMap<Chr, TrieNode>();

        is_end = false;
    }
}
```

abc

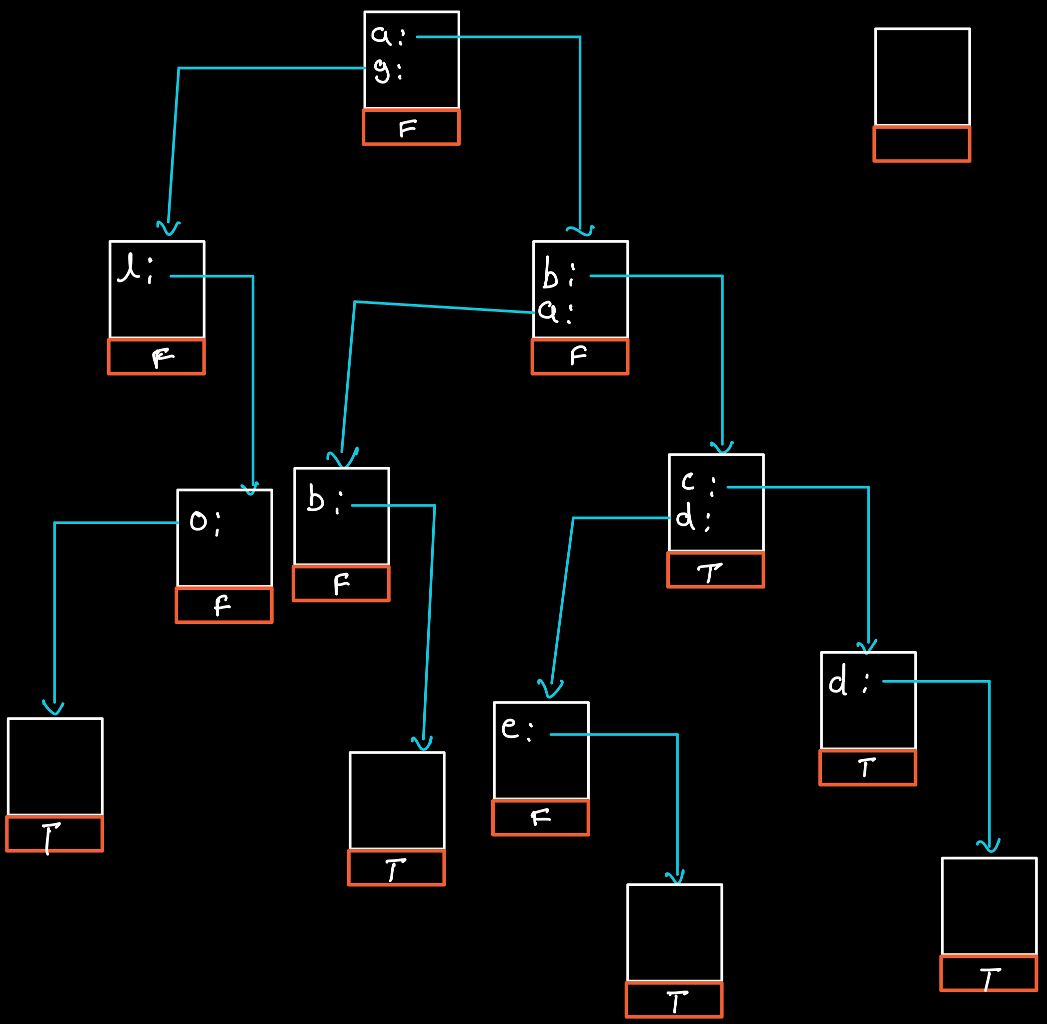
abde

abcd

aab

glo

ab



```
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.is_end = true;
}
```

TC: O(l)
SC: O(l)
l→ word.length()

```
boolean    search(word){

     TrieNode curr = root;
     for (i=0; i<word.length(); i++){
            ch = word.charAt(i);
            if( ! curr.map.containsKey(ch)){
                       ret false;
            }

            curr = curr.map.get(ch);
     }
     ret curr.is_end;
}
```

TC: O(l)

HW → Delete

Q. Given an array of words. Return an array of strings containing the smallest unique prefix for every word.
(Assume that unique prefixes will exist for every word in the array)

A : [cat, dog, rat, tiger, racoon]

[c, d, rat, t, rac]

A : [dog, doog, doe, donkey, duck, done]

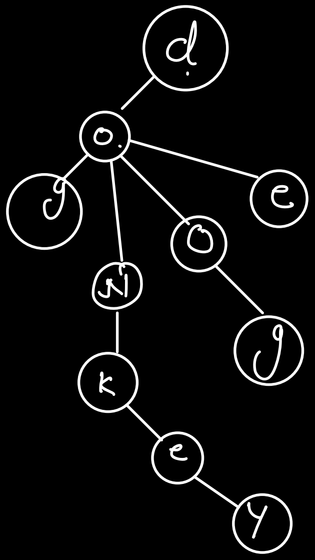
[dog, doo, doe, don, du]

dog → d

doog → doo

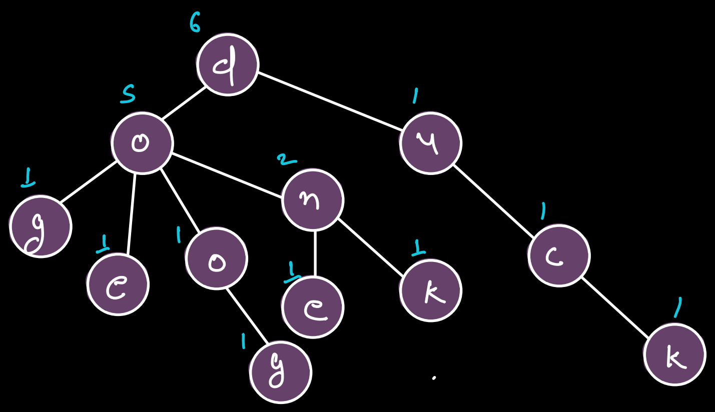
doe → doe

donkey → don



[dog, doog, doe , donk , duck, done]

[dog, doo, doe, donk, du, done]



Google

Given an array. Return the max XOR of any pair.

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

25^ 5 ⟹ 28

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

3^4

6^1 > 7

2^5

Brute Force

iterate over all pairs $O(N^2)$

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

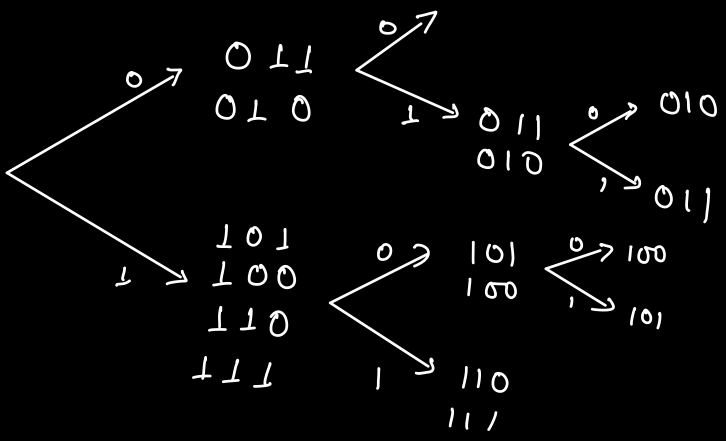
max XOR with 1 : O(N)
max XOR with 2 : O(N)
max XOR with 3 : O(N)
⋮

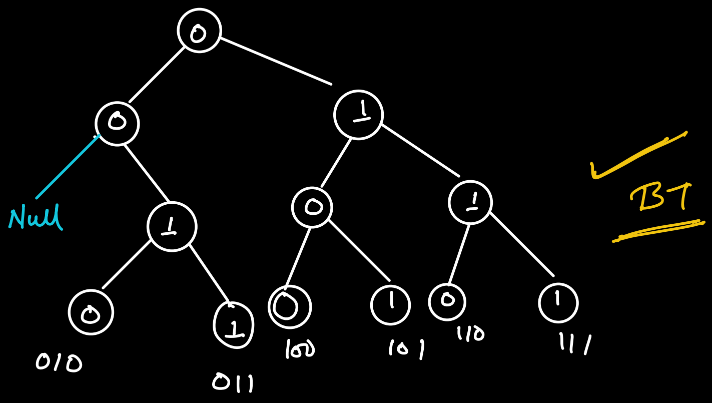
maxXOR

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

A[i]		No that gives Max XOR with A[i]
5	101	2
3	011	4
4	100	3
6	110	3
7	111	2
2	010	5

$1 \wedge 0$
$0 \wedge 1$ > 1





→ Build a trie using the binary representation of all no.
(from MSB to LSB) $O(N \times \log Max)$

→ Iterate over the array to find the max XOR for the curr array element. $O(N \times \log Max)$

↳ Move towards opposite value in trie.

TC : $O(N \times \log Max) \,/\, O(N\underset{\uparrow}{b})$

No of bits in datatype