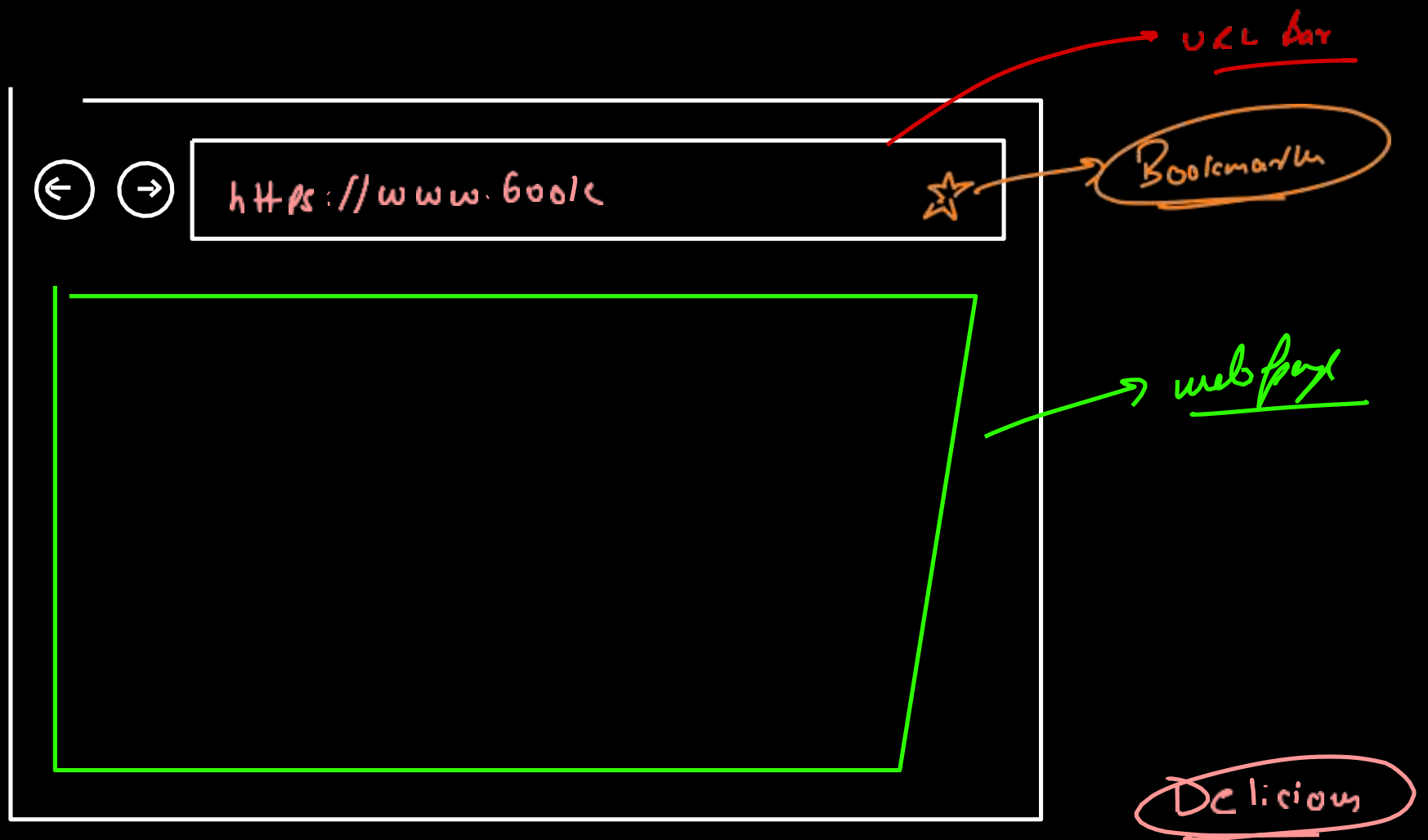




Backtracking 03 Tries





Assume that we have to prepare a bookmark manager

↳ Store a new bookmark ←

↳ Search for bookmarks →

Q what kind of data we have in the bookmark??

→ string

small alphabets

0-9

/ - _ & ! : = ; ...

→ <https://www.bookmyshow.com>

→ <https://www.bookmyshow.com/movies/avengers-end-game>

→ <https://www.leetcode.com/rearrange-string>

unordered-map / hashmap

→ problem 1 doesn't directly support
prefix searching

→ prob 2

2 strings might be having common prefixes, & hashmap will not utilize this property. & will individually store them.

→ x possible valid chars for making url

— — — — — — — — — — (n length)

↪ 1 byte → URL → — $n=1$ → x → x bytes

2 byte → URL → — — $n=2$ → x^2 → $2x^2$ bytes

3 byte → URL → — — — $n=3$ → x^3 → $3x^3$ bytes

⋮ ⋮ ⋮

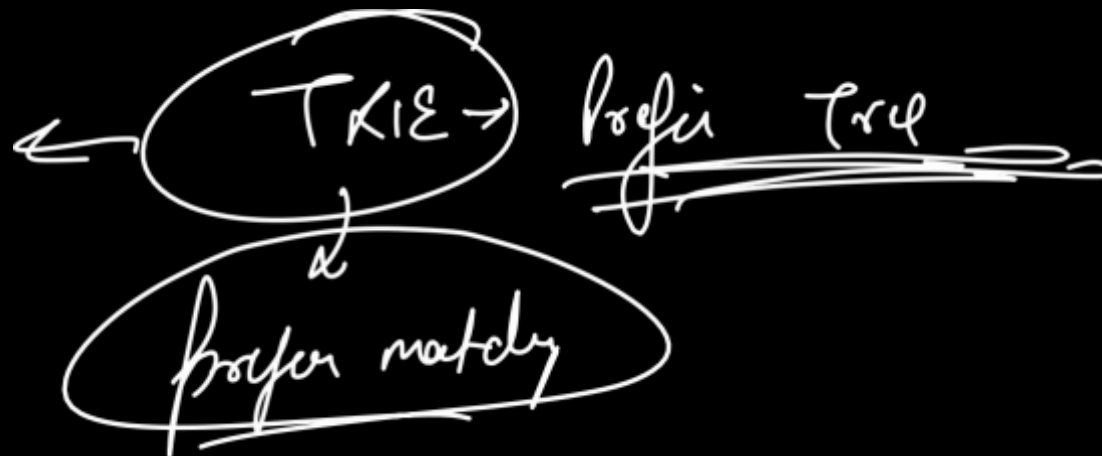
n byte → URL → — — — — — n → $n x^n$

$$\rightarrow x + 2x^2 + 3x^3 \dots \dots nx^n$$

↪

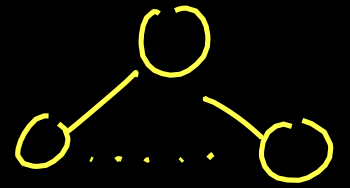
$$\text{Space} \rightarrow \underline{\underline{O(nx^n)}}$$

Contact list Sort Profs story



TRIE → Prefix tree

↳ n-ary
↳ generic

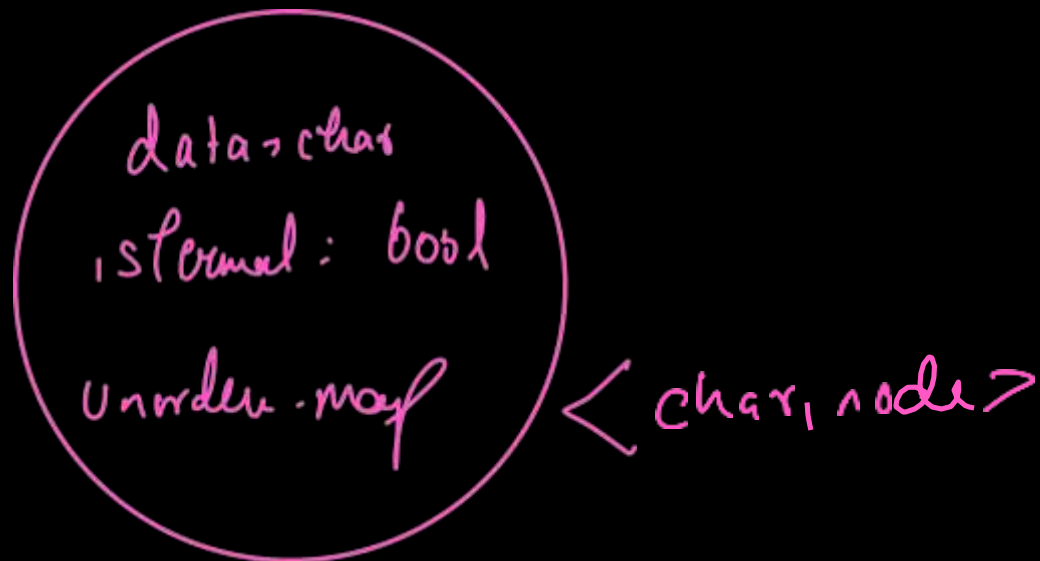


We make one node for each

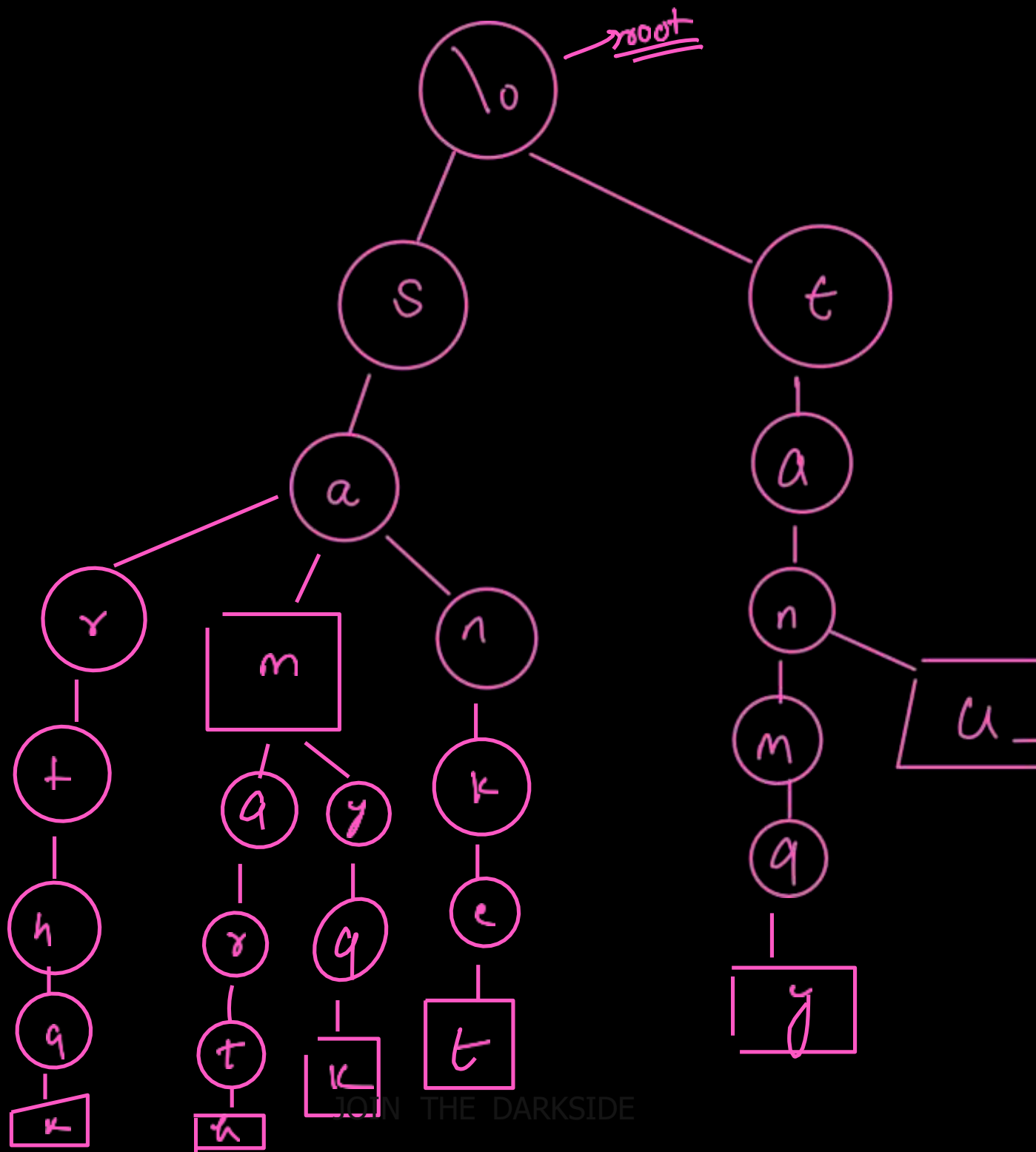
char. and if 2 strings have

common prefixes then we make single node

for them.



Sam
Sanket
Sarthak
Samarth
Samyak
Tammy
Tanu
→

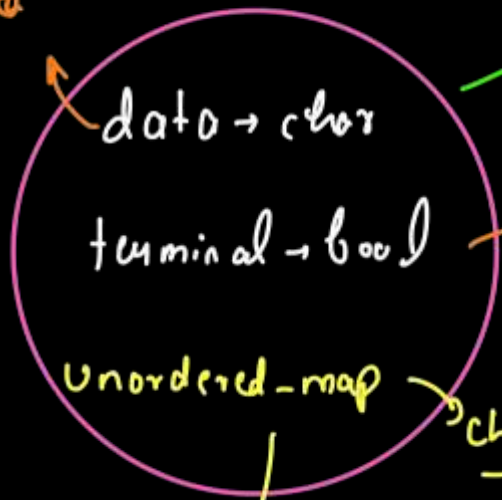


2

$$\rightarrow x + x^2 + x^3 - \dots - x^n$$

$$O(x^n)$$

data is to be represented by node



→ Tree Node

is the current node representing
end of a char.

n-ary

→ children

<key, value>

↓
char

↳ Tree Node

↳ address of
node

data
of child

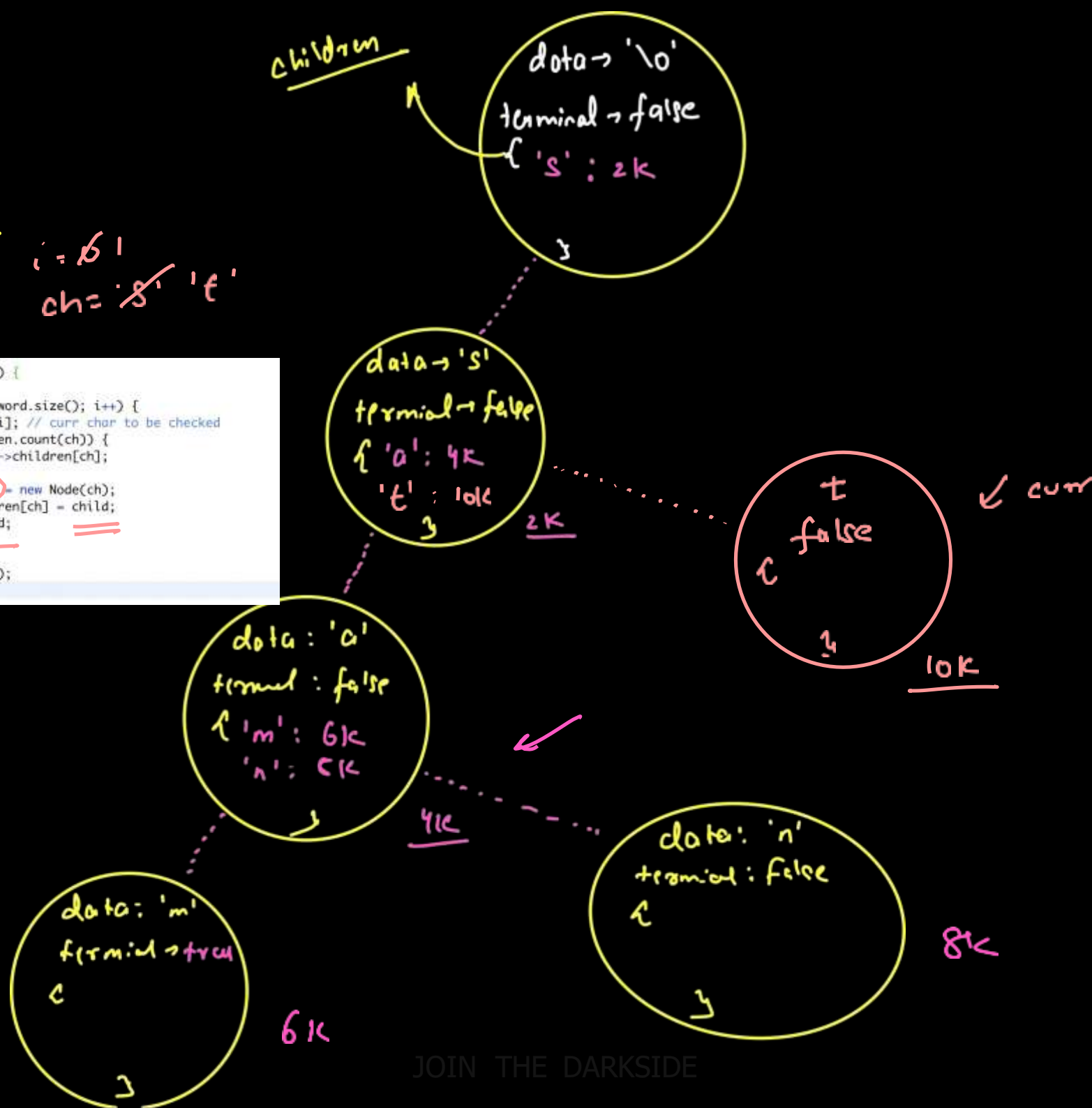
Sam ✓
Samuel

Samyak

St

i = 6
ch = 's' 't'

```
void insert(string word) {  
    Node* curr = root;  
    for(int i = 0; i < word.size(); i++) {  
        char ch = word[i]; // curr char to be checked  
        if(curr->children.count(ch)) {  
            curr = curr->children[ch];  
        } else {  
            Node* child = new Node(ch);  
            curr->children[ch] = child;  
            curr = child;  
        }  
    }  
    curr->makeTerminal();  
}
```



$$\leq 2^{31} - 1$$

2 2 2 2 2

25

nums →

[3, 10, 5, 25, 2, 8]

$n \leq 10^5$

3, 10

10, 5

5, 25

3, 5

10, 25

5, 2

3, 25

10, 2

5, 8

3, 2

10, 8

3, 8

25, 2

2, 8

25, 8

Max xor pair

001101
001100
000011
000000
000001
000010

Brute force

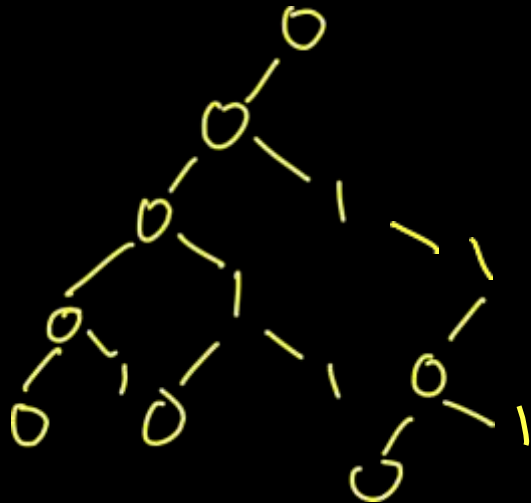
↓
generate all possible

pair

$O(n^2)$

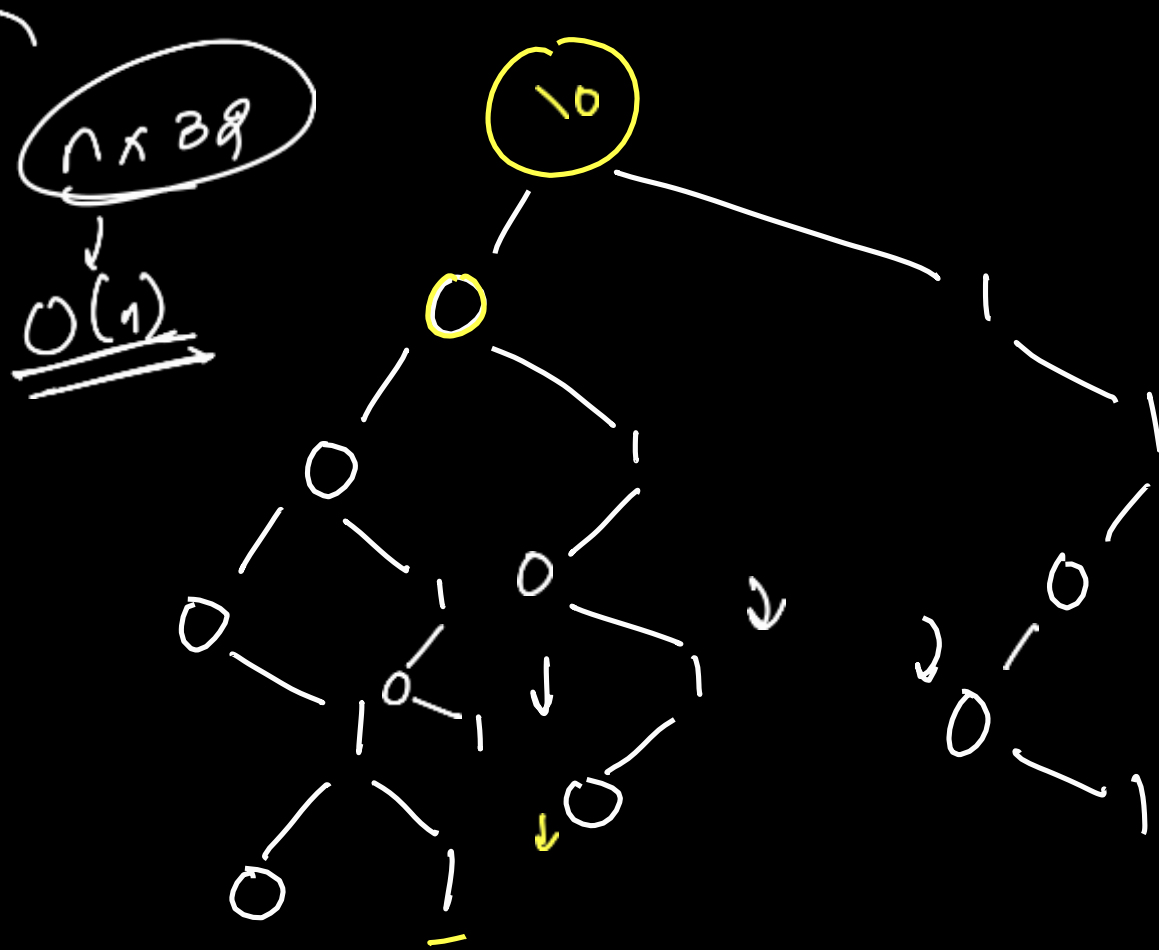
Binary Trie

→ using 0's & 1's



$3 \rightarrow 00011$
 $10 \rightarrow 01010$
 $5 \rightarrow 00101$
 $25 \rightarrow 11001$
 $2 \rightarrow 00010$

11100 $\rightarrow 28$



0 0 1 0
 0 1 1 1
 1 0 1 1
 1 1 1 0

1 1 1 1
 1 0 1 → 5 12
 0 1 0 =
 1 1 1
7

1 1 1 1
 1 0 1 → 5 12
 1 0 1
 0 0 0
 0

1 1 1 1
 0 1 0 1
 0 0 1 0
 0 1 1 1
 2
7

1 1 1 1
 0 1 0 1
 1 1 0 1
 1 0 0 0
 8

To maximize a no, we should make bits as different as possible given most significant bit given priority.



▶ **THANK YOU** ◀