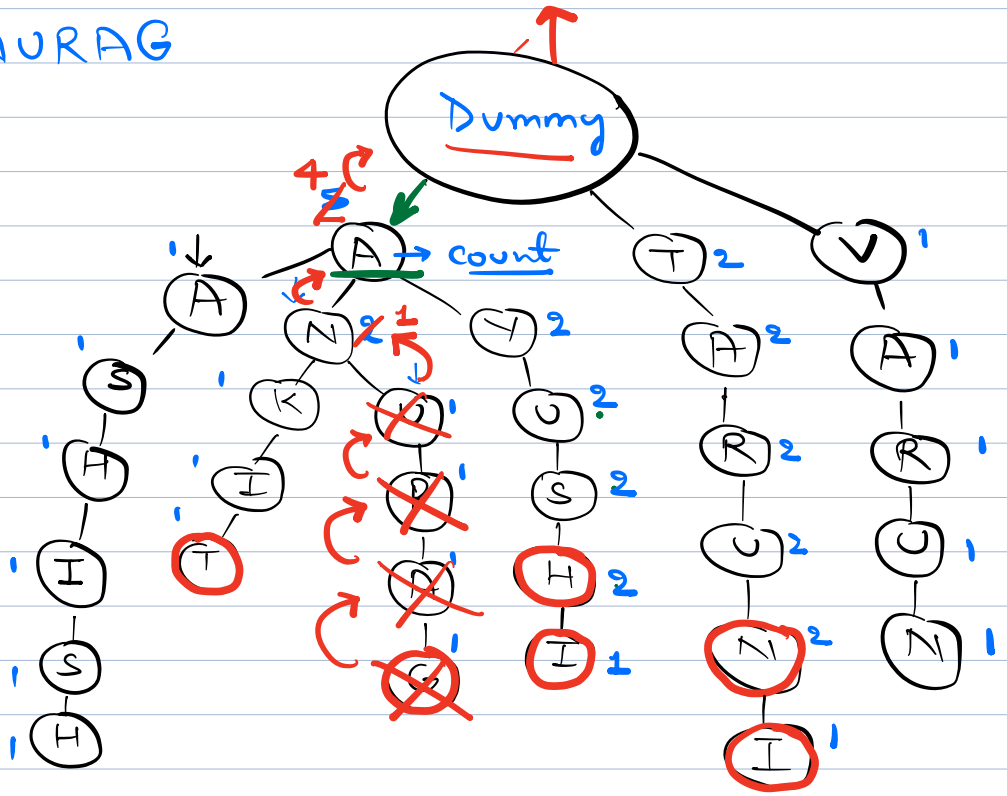
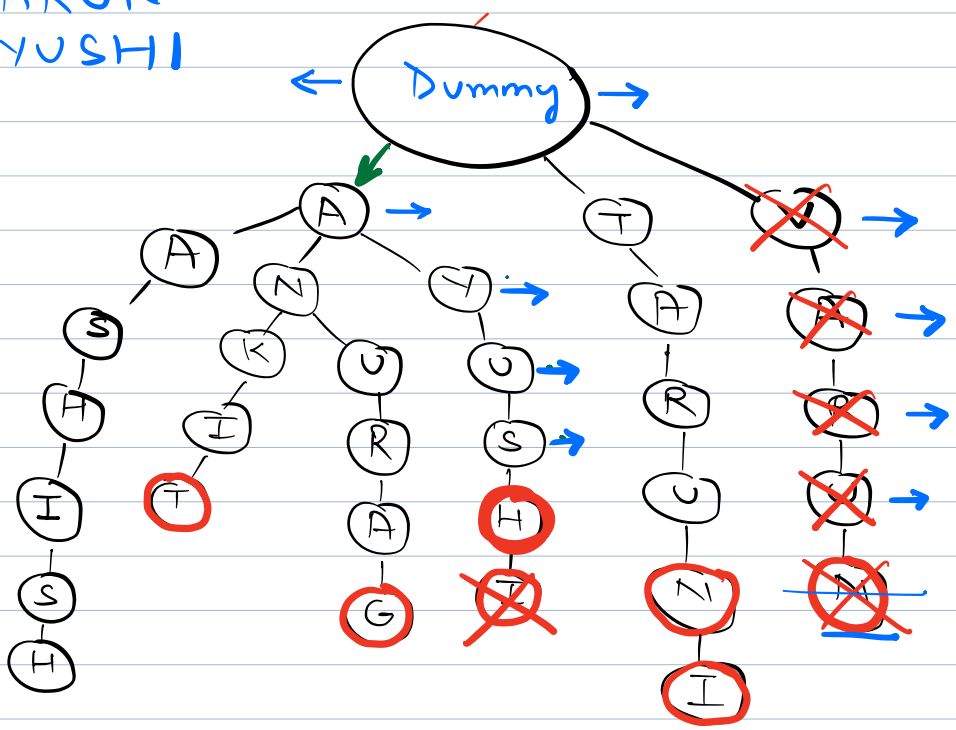
ANURAG



~~1)~~ VARUN

~~2)~~ AYUSHI



Code

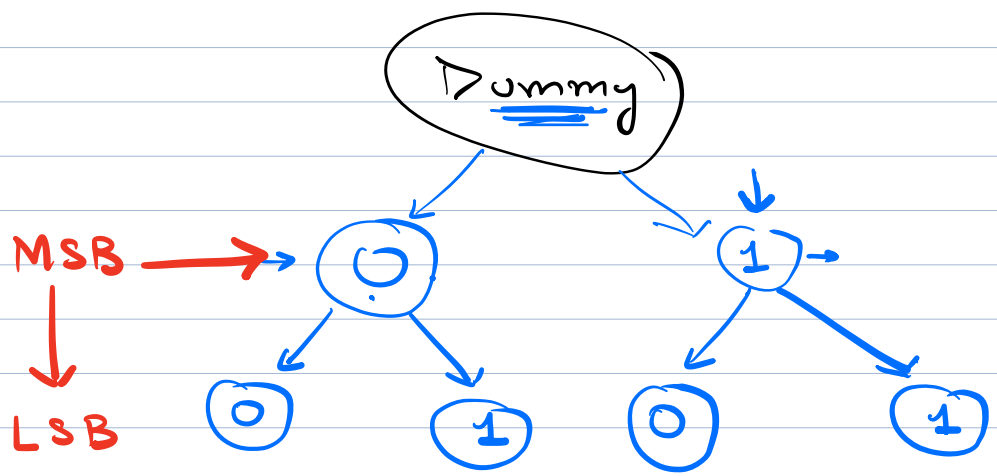
```
Node {
    char c
    HM<char, Node> children
    boolean isEnd;
}

boolean delete (root, word, index) {
    if (index == word.length - 1) {
        if (root.children.isEmpty()) {
            free (root); (
            return true;
        }
        else {
            root.isEnd = false;
            return false;
        }
    }
    char nextChar = word.charAt(
                    index);
    Node nextNode = root.children.
                    get (nextChar);
    boolean isNextCharDeleted =
            delete (nextNode, word, index+1);
    if (isNextCharDeleted == True) {
```

```
root.children.delete(nextChar);
if (root.children.isEmpty()
    && root.isEnd == false) {
        free(root);
        return true;
    }
}
return false;
}
```

Trie Of Bits (Binary Tree)

00, 01, 10, 11



Given an array of integers. (A)
find the max value of

$A[i] \wedge A[j] \quad \forall\ i, j.$

Ex: $[9, 8, 10, 7]$

$9 \wedge 8 = 1$
$9 \wedge 10 = 3$
$9 \wedge 7 = 14$
$8 \wedge 10 = 2$
Ans $8 \wedge 7 = 15$
$10 \wedge 7 = 13$

1) Brute force

$O(N^2 \times 32) = O(N^2 \log N)$

($32 \downarrow \log N$)

2) $1000 > 0111$

(I) (II)

MSB overpowers.

x	y	x ^ y
0	0	0
0	1	1
1	0	1

1 1 0

N no's

1 BN

1	0	0	0	1
5	0	1	0	1
3	0	0	1	1
8	1	0	0	0
7	0	1	1	1
6	0	1	1	0

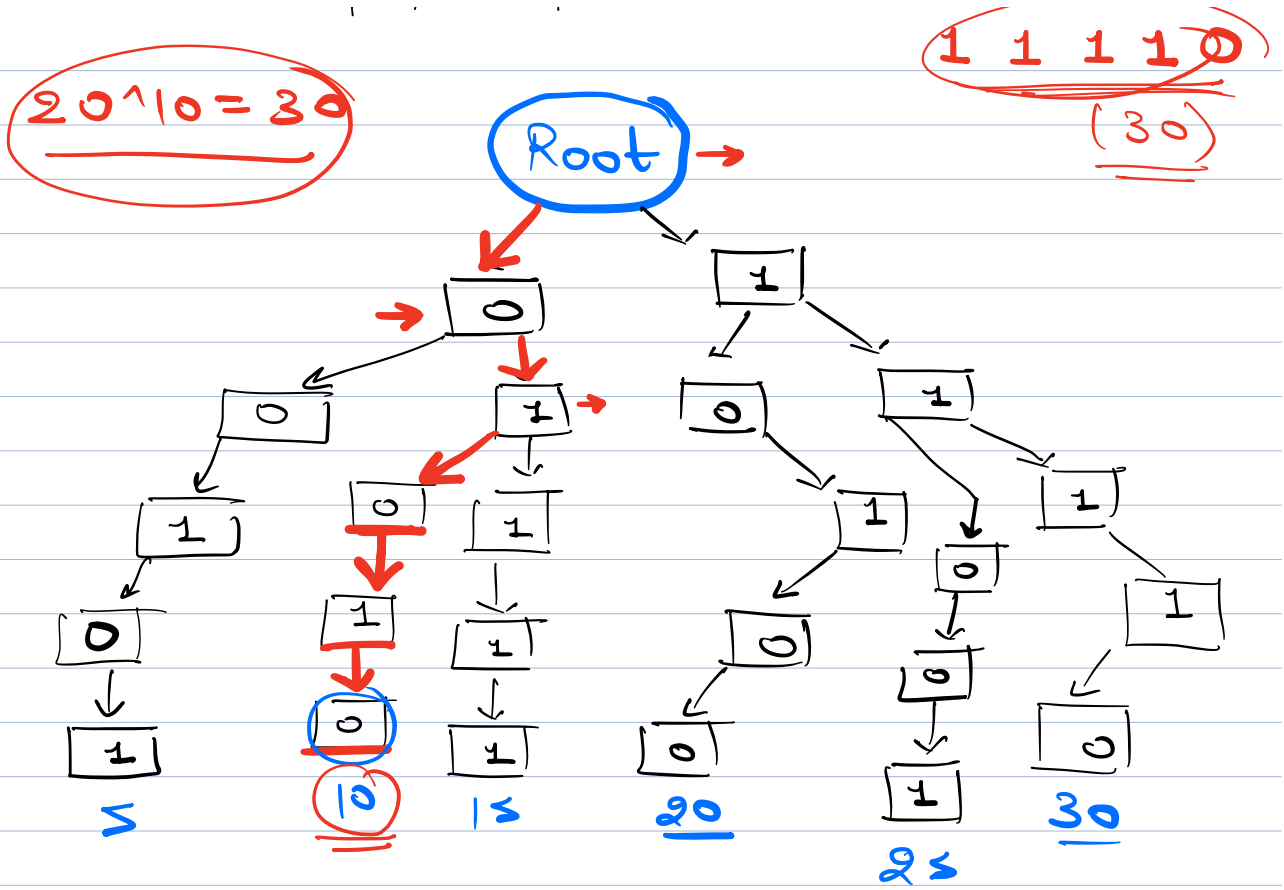
2 → 0 0 1 0

MSB ⟶ LSB

2^x = 1 _ _ _

A: [20, 30, 15, 25, 10, 5] K=5

20 :	1	0	1	0	0	^20 = 0
30 :	1	1	1	1	0	^20 = 16
15 :	0	1	1	1	1	^20 = 27
25 :	1	1	0	0	1	^20
10 :	0	1	0	1	0	^20 = 30
5 :	0	0	1	0	1	^20



T.C. to create try.

$= O(N \times 32)$

$\approx (N \log N)$

T.C. $= O(32) \approx O(\log N)$

T.C. $= O(N \log N)$

Code : (H.W.)

Q Given an array of integers. Find max subarray XOR value.

A: [1, 4, 2, 6] (K = 3) Assume 3 bits

1 :	0	0	1
4 :	1	0	0
2 :	0	1	0
6 :	1	1	0

[1] → 1
[4] → 4
[2] → 2
[6] → 6
[1,4] → 5
[4,2] → 6

[2,6] → 4
[1,4,2] → 7
[4,2,6] → 0

[1,4,2,6] → 1

Ans

1) Brute force

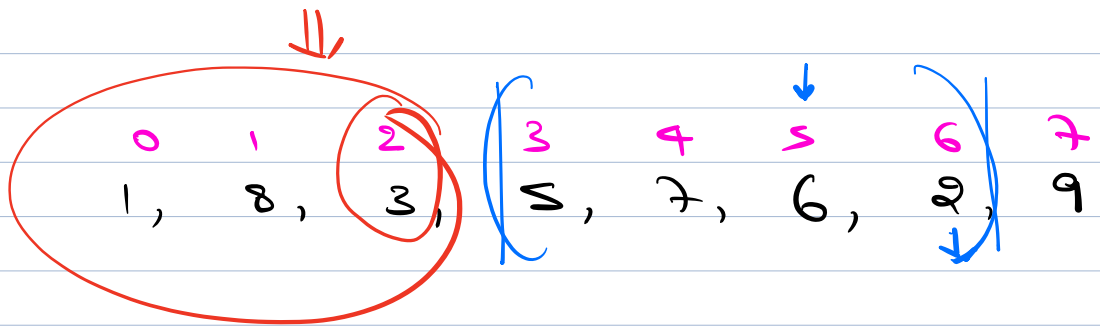
```
ans = 0;
for (i = 0; i < N; i++) {
```

e

```
for (j = i ; j < N ; j++) {
```

// find XOR of all elements from i to j

1) Loop
2) Prefix XOR



(5 ^ 7 ^ 6 ^ 2)

Pre[i] = XOR of all nos from 0 to i

Prefix[6] = (1 ^ 8 ^ 3 ^ 5 ^ 7 ^ 6 ^ 2)

Prefix[2] = (1 ^ 8 ^ 3)

XOR [3, 6] = Prefix [6] ^ Prefix [2]

XOR [s, e] = Prefix [e] ^ Prefix [s-1]

$\forall (s, e)$ max

Soln 1) Create prefix XOR array

2) Use exact soln of above problem to find the max

[a, b, c, d, e]

0 1 2 3 4

Pre [a, a^b, a^b^c, a^b^c^d, a^b^c^d^e]

Pre[4] ^ Pre[1]

(a^b^c^d^e) ^ (a^b)