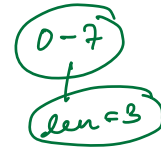
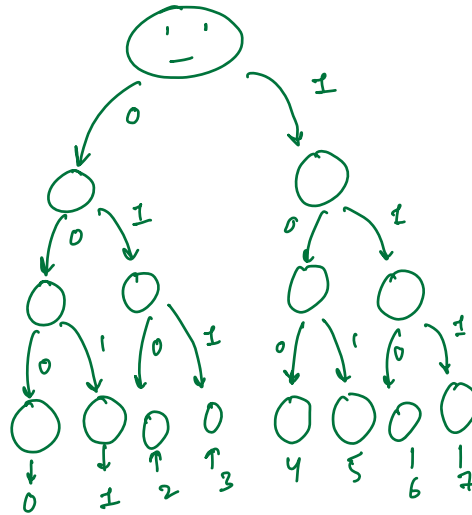


Trie on Bits \rightarrow Binary tree

de-show



MSB



0-31
height of tree/trie
 $h=5$

0-2^h-1

32
0-2³²-1

```
class Node {
    // data
    Node children[2];
}

children[0], children[1];
```

Node 200
one;

Find max value of $A[i] \oplus A[j]$ for pairs (i, j)

arr: [9 8 10 7]

$8 \oplus 7 = 15$

9: 1001

$9 \oplus 8 = 0001 = 1$

8: 1000

$8 \oplus 7 = 1111 = 15$

10: 1010

7: 0111

Xor

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

B.F:

consider all possible pairs $\rightarrow O(n^2)$

maximize xor

$15 = 1111$

1 0 0 0
↓ ↓
0

$7 = 0111$

$11 = 1011$

$12 = 1100$

$14 = 1110$

Maximize xor try setting more significant bits

```
ans = max(ans, arr[i] * elset);
```

5 bits

Handwritten notes showing a sequence of numbers: 5, 5, 10, 5, 5, with a large green oval around the number 10.

10101

10000

01011

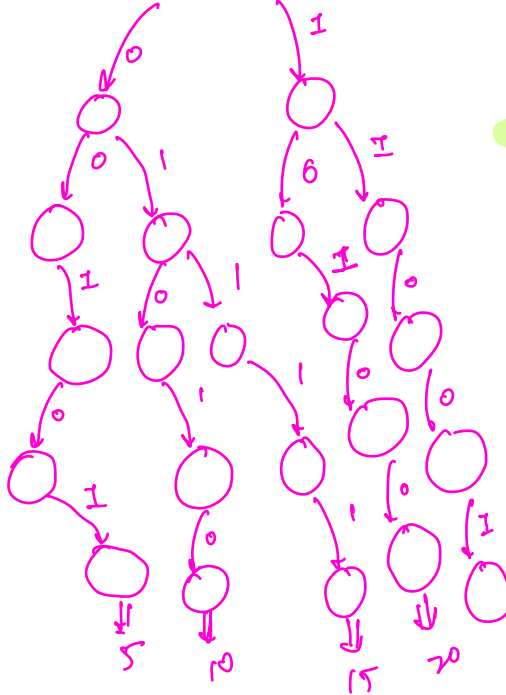
01010 = 10

७७ ११०

0001

AD 0101

11011


$$15^5 = 10$$
$$15^{10} = 5$$

1) queue $\rightarrow O(n)$

2) moet

$$v_0(r)$$

$h = \text{no of bits}$

T.C: $n \times \text{max no of bits}$

S.C: 2 max no of lab

2

1
2
4
8
16
32
64
128
256

Maximum subarray xor

[1, 4, 3]

$$[1] = 1$$

$$[4] = 4$$

$$[3] = 3$$

$$[1, 4] = 5$$

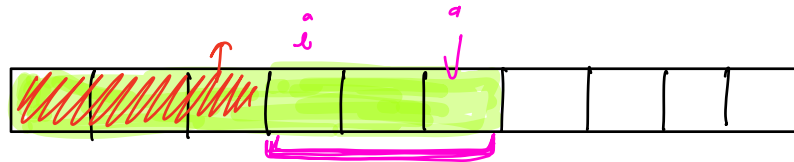
$$[4, 2] = 7$$

$$[1, 4, 3] = 6$$

B.F: consider all possible subarrays

$$\begin{matrix} a \wedge b \wedge c \wedge d \\ \text{xor} \left\{ \begin{matrix} a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge g \wedge h \end{matrix} \right. \end{matrix}$$

prefixor



$$\text{prefixor}(i) = \text{xor from } 0 \text{ to } i$$

$$\text{prefixor}(j) \wedge \text{prefixor}(i-1) = \text{xor}(i \text{ to } j)$$

$$2 \quad 4 \quad 6 \quad 9 \quad 1 \quad 5 \quad 7 \quad 4$$

$$2 \wedge 4 \wedge 6 \wedge 9 \wedge 1 \wedge 5$$

$$2 \wedge 4 \wedge 6$$

$$\Rightarrow 2 \quad 2^4 \quad 2^4 6 \quad 2^4 6^1 \quad 2^4 6^1 9$$

pf

	2	4	6	1	9	8
	2	6	12	13	22	30

prefixxor = [1 5 6]

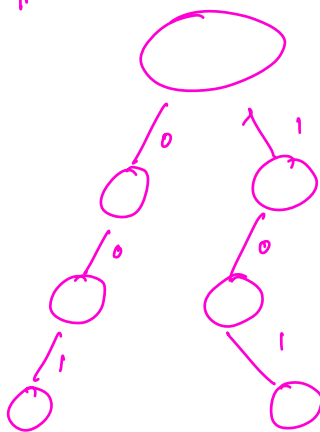
ans = 6

ans = max(6, 5 ^ 1)

ans = max(6, 6 ^ 1)

5 = 101

6 = 110



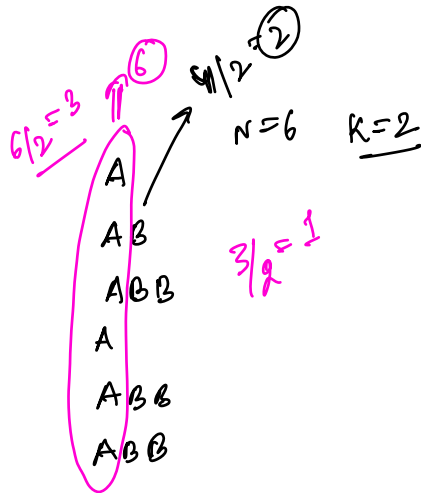
Bundle strips (Google Kickstart)

N strips \equiv divide into groups of size K

$$n \div k = 0$$

Score \equiv len of longest common prefix
of that group

Maximum total score of all groups

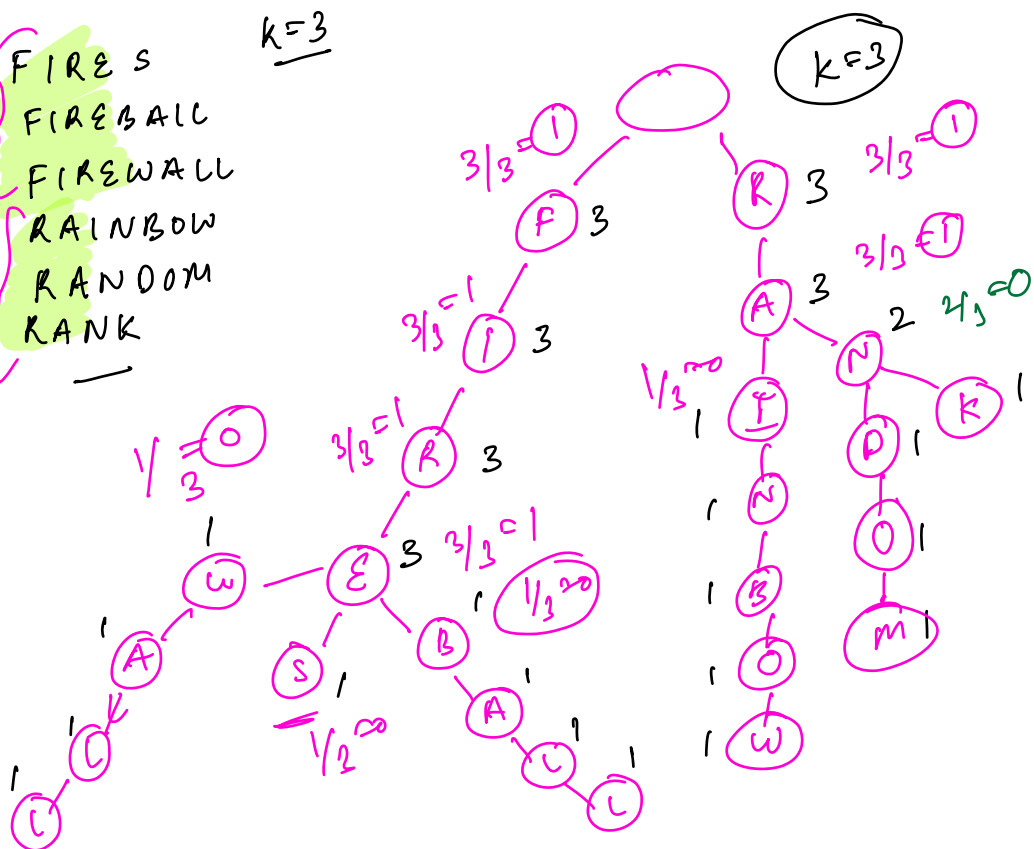


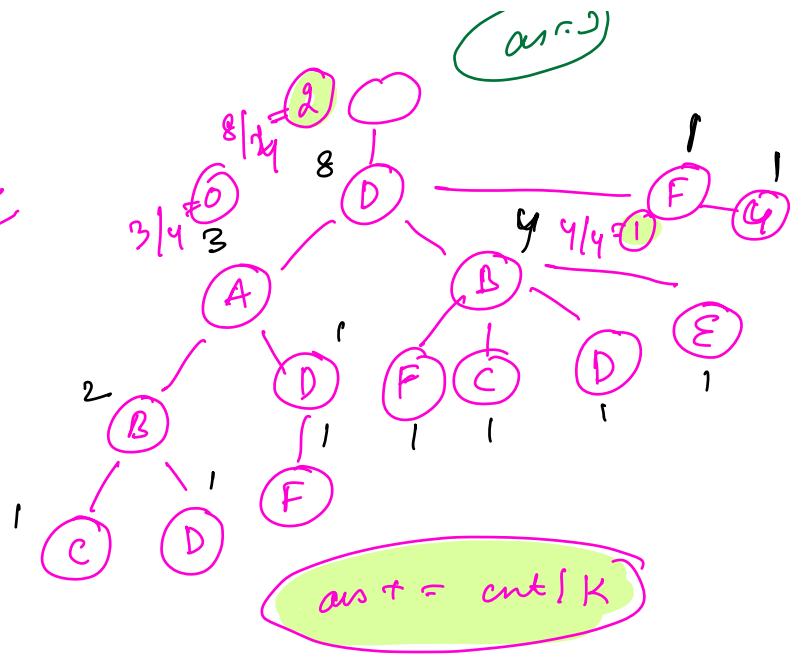
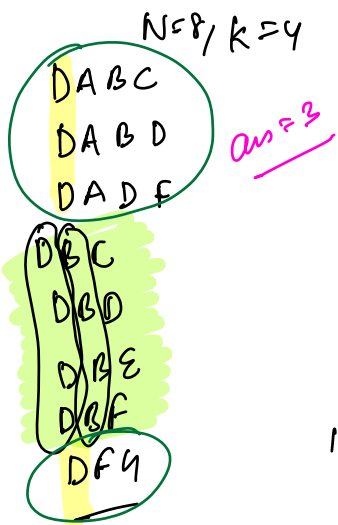
2	2	2
<u>G₁</u>	<u>G₂</u>	<u>G₃</u>
A	AB	ABB
A	ABB	ABB
\Downarrow	\Downarrow	\Downarrow
1	2	3
A	A	ABB
AB	ABB	ABB
<hr/>	<hr/>	<hr/>
1	1	3
<hr/>	<hr/>	<hr/>

$\textcircled{6}$

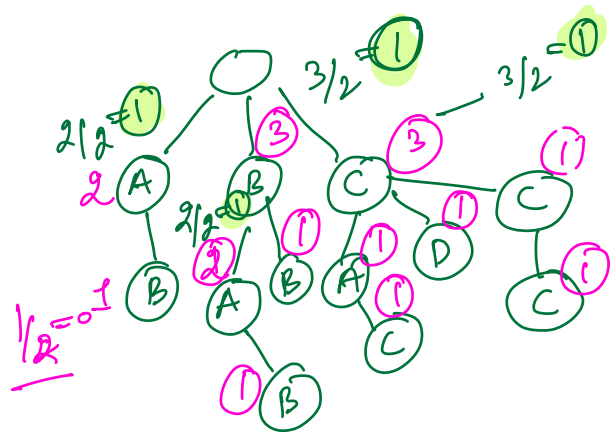
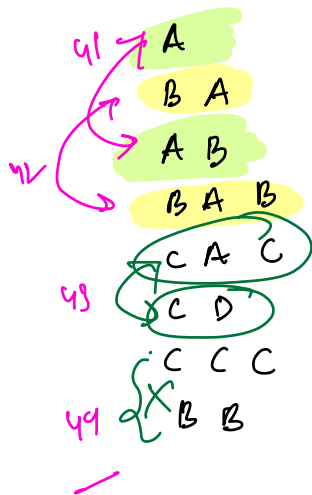
FIRE S
 FIREBALL
 FIREWALL
 RAINBOW
 RANDOM
 RANK

k=3





$K=2$



```

void visit (node root, int k)
{
    if (root == null) return;

    ans += cnt / k;

    for (i=0; i<26; i++)
    {
        if (root.children[i].cnt >= k)
            visit (root.children[i], k);
    }
}

```