

video-78

# GRAPHS...

"let's make it easy too"

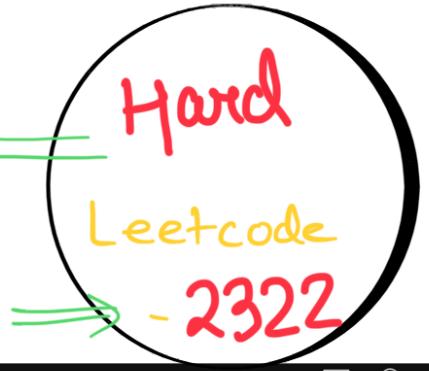
If you have tried my  
"Graph Concepts & Qns" playlist,  
these Qns, will seem very easy.  
Do try it once :)



Facebook  
Instagram] → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK → 



Weekend



## (M)otivation :-

There are no limits to what you can achieve through dedication...



MIK..

### 2322. Minimum Score After Removals on a Tree

Hard Topics Companies Hint

There is an undirected connected tree with  $n$  nodes labeled from 0 to  $n - 1$  and  $n - 1$  edges.

You are given a 0-indexed integer array  $\text{nums}$  of length  $n$  where  $\text{nums}[i]$  represents the value of the  $i^{\text{th}}$  node. You are also given a 2D integer array  $\text{edges}$  of length  $n - 1$  where  $\text{edges}[i] = [a_i, b_i]$  indicates that there is an edge between nodes  $a_i$  and  $b_i$  in the tree.

Remove two distinct edges of the tree to form three connected components. For a pair of removed edges, the following steps are defined:

1. Get the XOR of all the values of the nodes for each of the three components respectively.
  2. The difference between the largest XOR value and the smallest XOR value is the score of the pair.
- For example, say the three components have the node values: [4, 5, 7], [1, 9], and [3, 3, 3]. The three XOR values are  $4 \wedge 5 \wedge 7 = 6$ ,  $1 \wedge 9 = 8$ , and  $3 \wedge 3 \wedge 3 = 3$ . The largest XOR value is

Graph.

$n$  nodes,  
 $n-1$  edges

8 and the smallest XOR value is 3. The score is then  $8 - 3 = 5$ .

Return the **minimum** score of any possible pair of edge removals on the given tree.

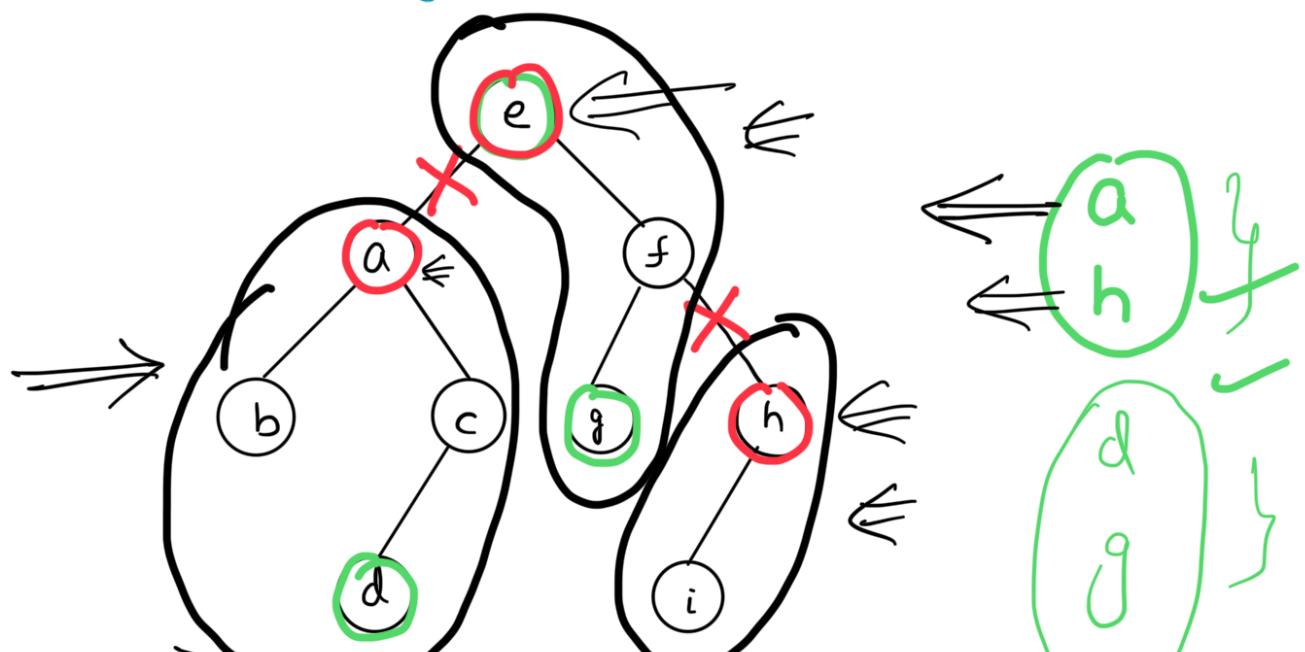
**Example 1:**

**Input:** nums = [1, 5, 5, 4, 11], edges = [[0,1], [1,2], [1,3], [3,4]]  
**Output:** 9

**Explanation:** The diagram above shows a way to make a pair of removals.  
- The 1<sup>st</sup> component has nodes [1,3,4] with values [5,4,11]. Its XOR value is  $5 \wedge 4 \wedge 11 = 10$ .  
- The 2<sup>nd</sup> component has node [0] with value [1]. Its XOR value is  $1 = 1$ .  
- The 3<sup>rd</sup> component has node [2] with value [5]. Its XOR value is  $5 = 5$ .  
The score is the difference between the largest and smallest XOR value which is  $10 - 1 = 9$ .  
It can be shown that no other pair of removals will obtain a smaller score than 9.

$10 - 1 = 9$

# Thought Process



Subtree

DFS(a)  $a \wedge b \wedge c \wedge d$

DFS(e)  $e \wedge f \wedge g$

DFS(h)  $h \wedge i$

```
for (node1 : ...) {  
    for (node2 : ...) {
```

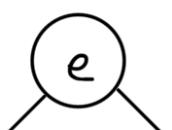
DFS(a)  $a \wedge b \wedge c \wedge d$

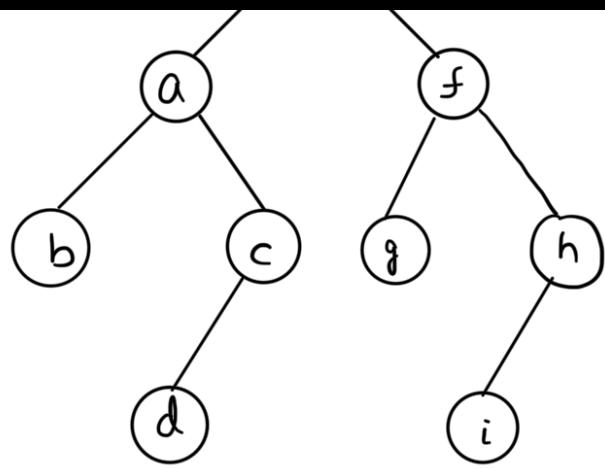
DFS(e)  $e \wedge f \wedge g$

DFS(h)  $h \wedge i$

} XOR.

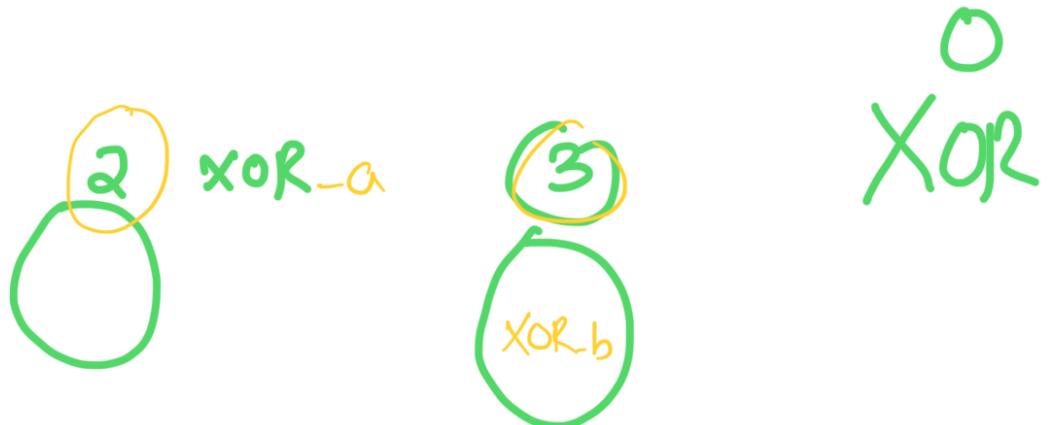
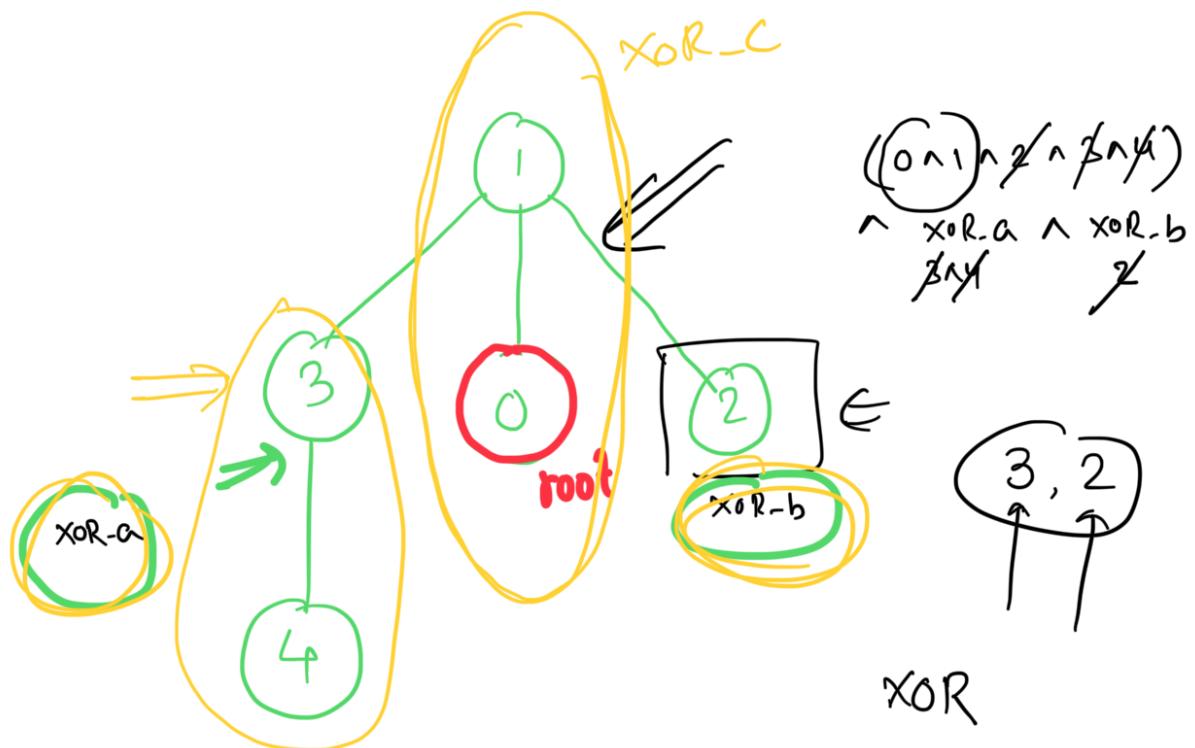
How to compute XOR  
of all possible  
Subtrees



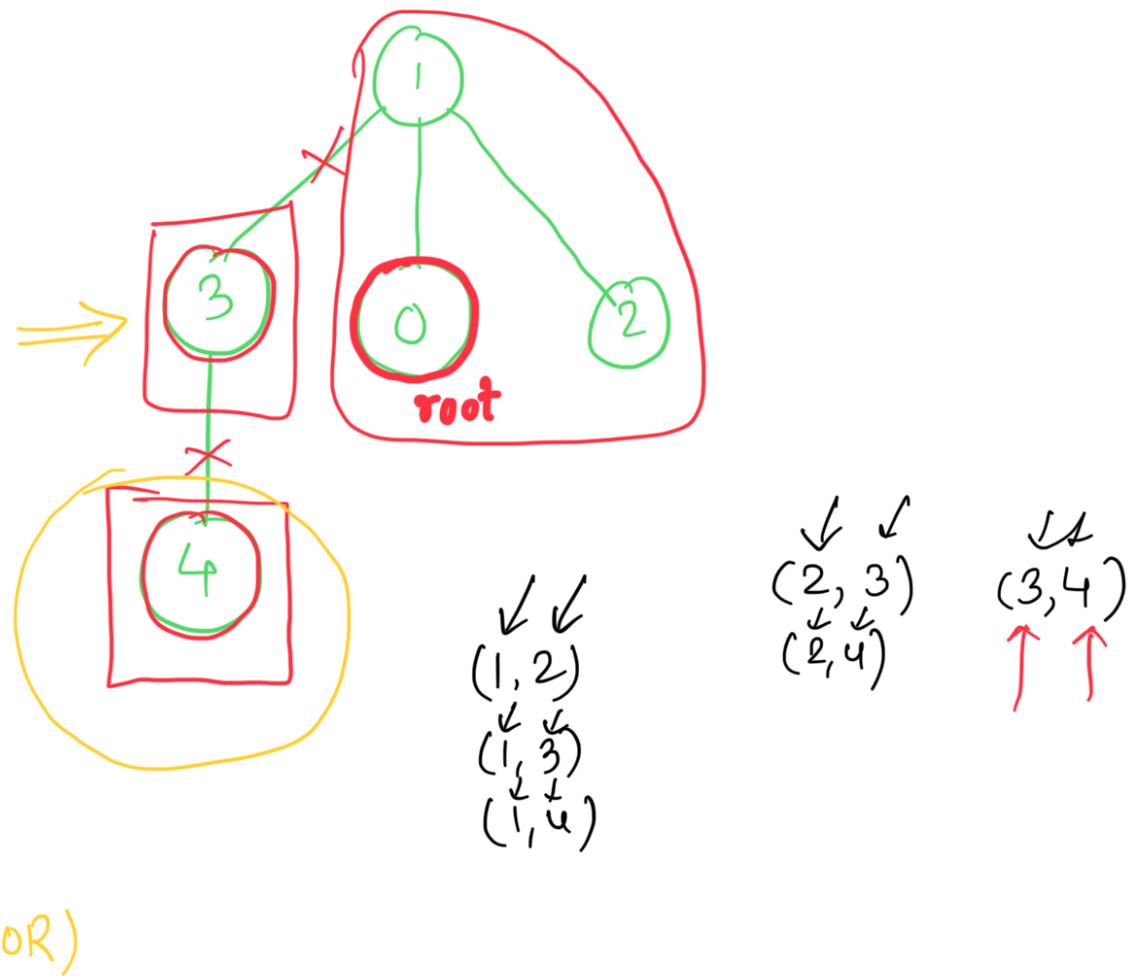


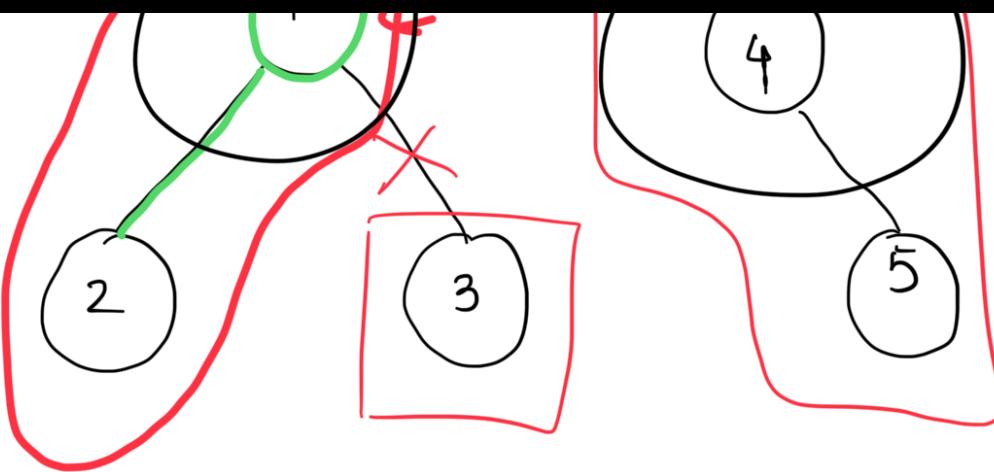
a, h

$\text{xor}[a] \mid \text{xor}[h]$



$$\text{xor}_c = \text{xor} \wedge \text{xor\_a} \wedge \text{xor\_b}$$





\*  $(1, 3) \Rightarrow 1 \text{ is ancestor of } 3$

$\textcircled{1} \rightarrow (1 \wedge 2 \wedge 3)$  subtree

$\textcircled{3} \rightarrow (3)$  subtree

~~$\textcircled{1}$~~   $\Rightarrow (1 \wedge 2)$

~~$\textcircled{3}$~~   $\Rightarrow (3)$

$(XOR) \wedge (1 \wedge 2) \wedge (3) \Rightarrow$  3rd compa  
XOR.

XOR subtree [0]

for ( node1 = 1 ; )

for ( node2 = node1 + 1 ; )

if (node1 is ancestor of node2) {

} else if (node2 is ancestor of node1) {

} else {

,

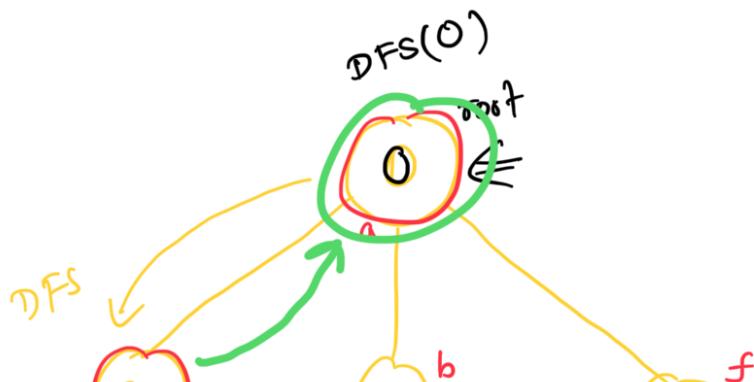


XOR ^ — ^ — ^ }

~~①~~ XOR of all subtrees.

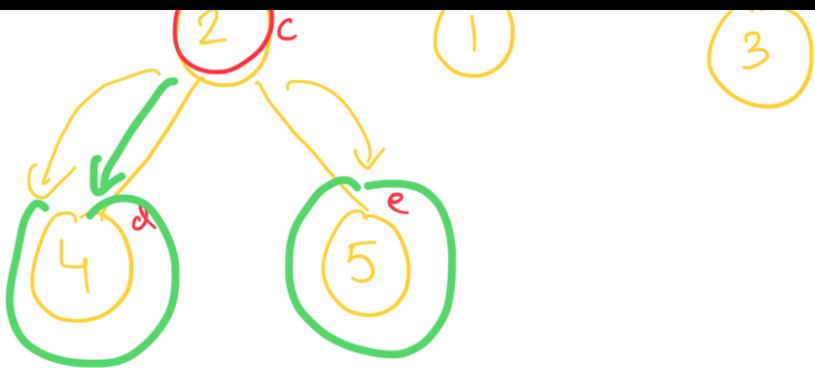
~~②~~ checking if (ancestor or not);

\* XOR of all subtrees



hums

[a | b | c | d] e } f



XorSubtree

0	1	2	3	4	5
a	Y	X	Z	0	0

$((a \wedge Y) \wedge X) \wedge Z$       DFS(0, -1);

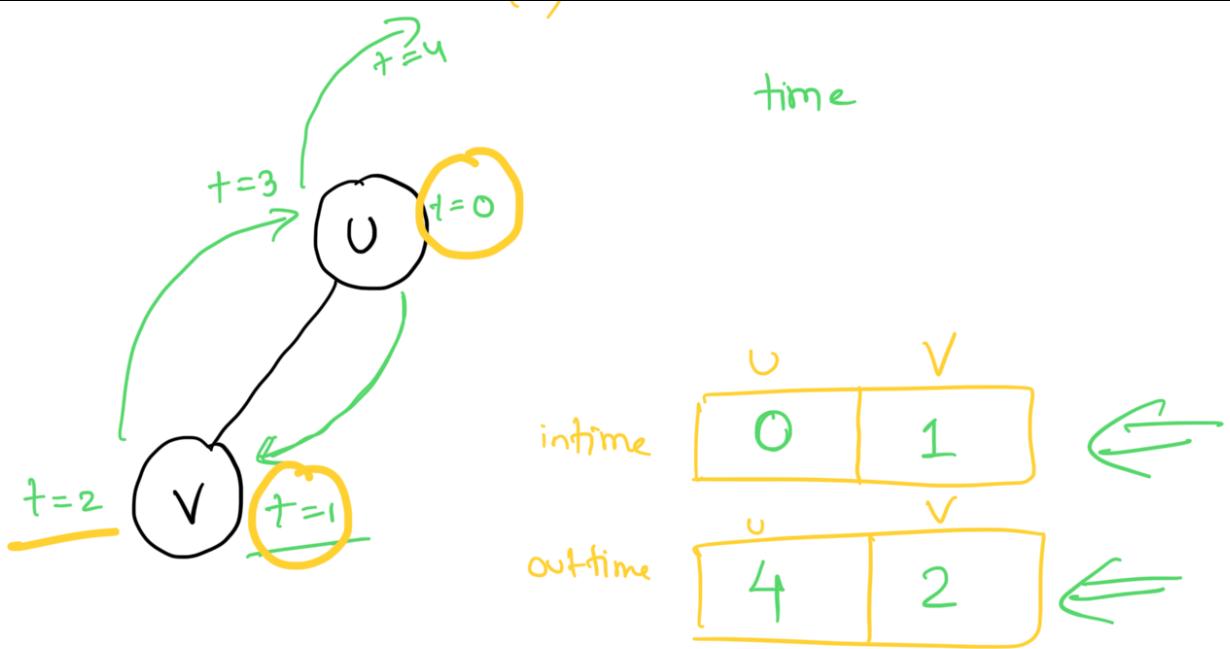
```

XorSubtree[node] = nums[node];
for (int & ngbr : adj[node]) {
    if (ngbr != parent) {
        dfs(ngbr, node);
        XorSubtree[node] ^= XorSubtree[ngbr];
    }
}

```

Checking if (ancestor or not)

(Check if a node is ancestor of another node) (v)



bool isDescendent( $U, V, \text{intime}, \text{outtime}$ );

{

$\text{intime}[V] \geq \text{intime}[U]$

$\text{outTime}[V] \leq \text{outTime}[U]$

$U$  is an ancestor of  $V$

$\Rightarrow (V$  is a descendant of  $U)$

XOR

Story points:-

treat 0 as root.

- ①  $2^n$  nodes pair  $\rightarrow$  all possible.
- ② XOR of all subtre...

~~order of all sources.~~

~~(3)~~

inTime & outTime  $\rightarrow$  Descendant  
check.



