

Trees 4

Revision

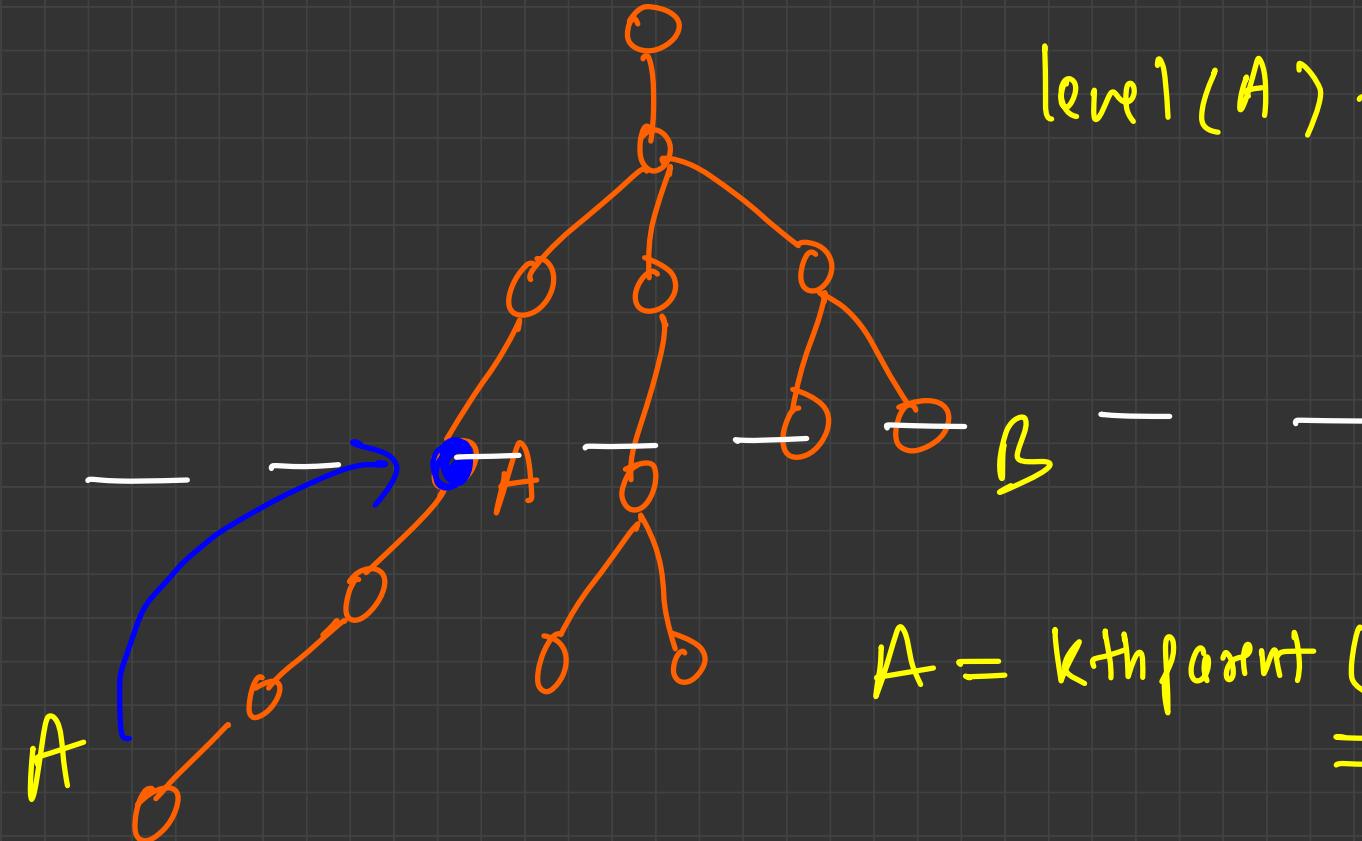
- properties
- Traversals
- In-time, out-time trick
- Binary lifting
- Diameter
- LCA \rightarrow $\Theta((\log n)^L)$

\hookrightarrow

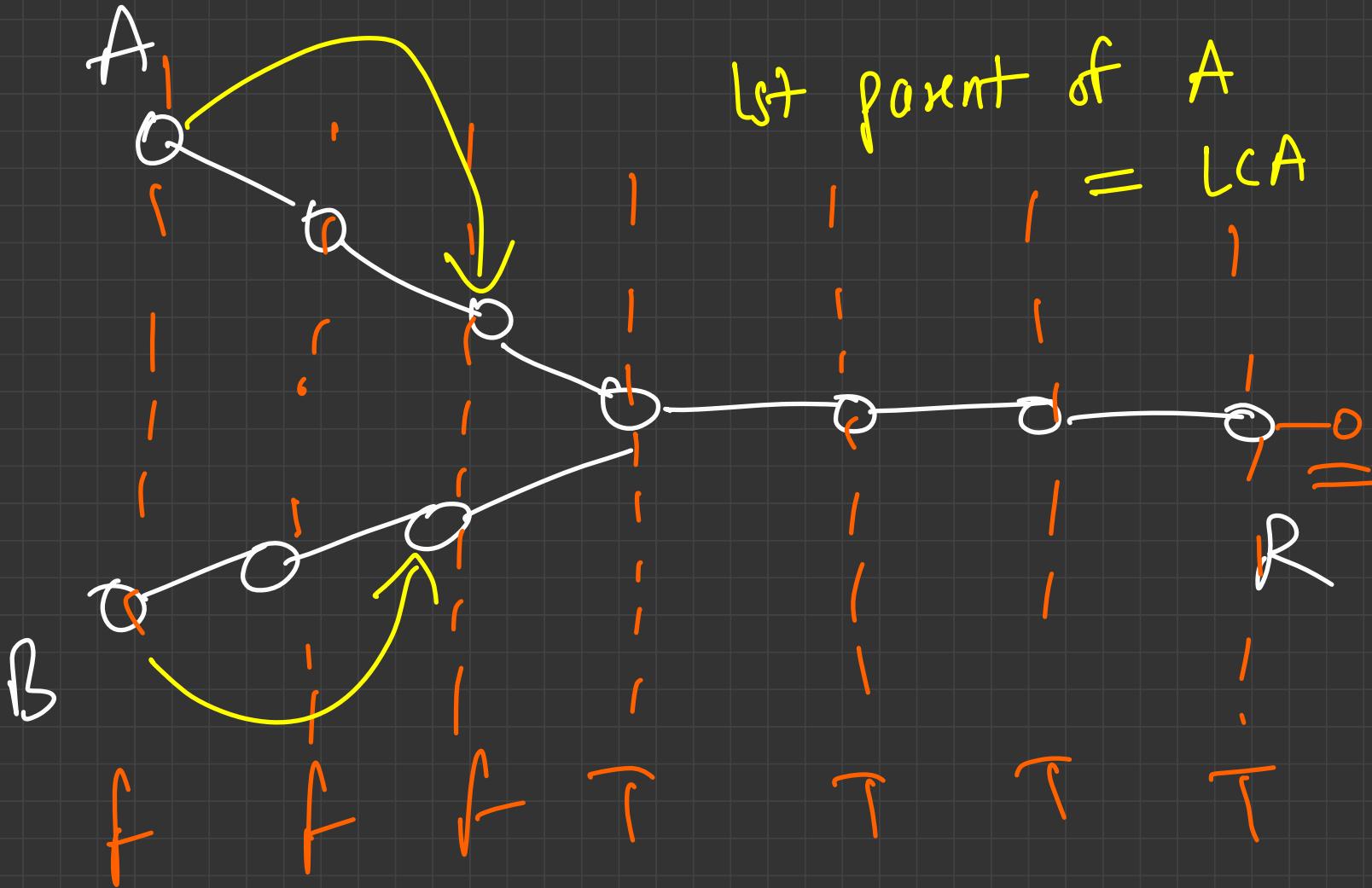
$O(\log n)$

$$\text{level}(A) - \text{level}(B)$$

$$= 3 \\ \underline{\underline{=}}$$

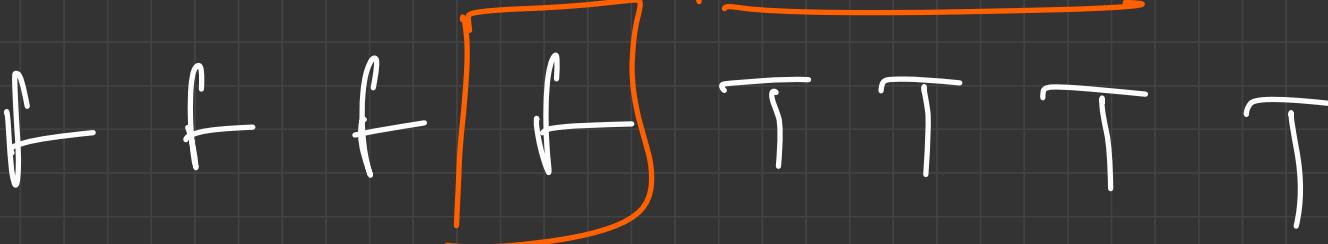


$$A = \underline{k\text{thparent}}(A, 3)$$



Binary search = $\underline{\underline{O(\log n)}}$

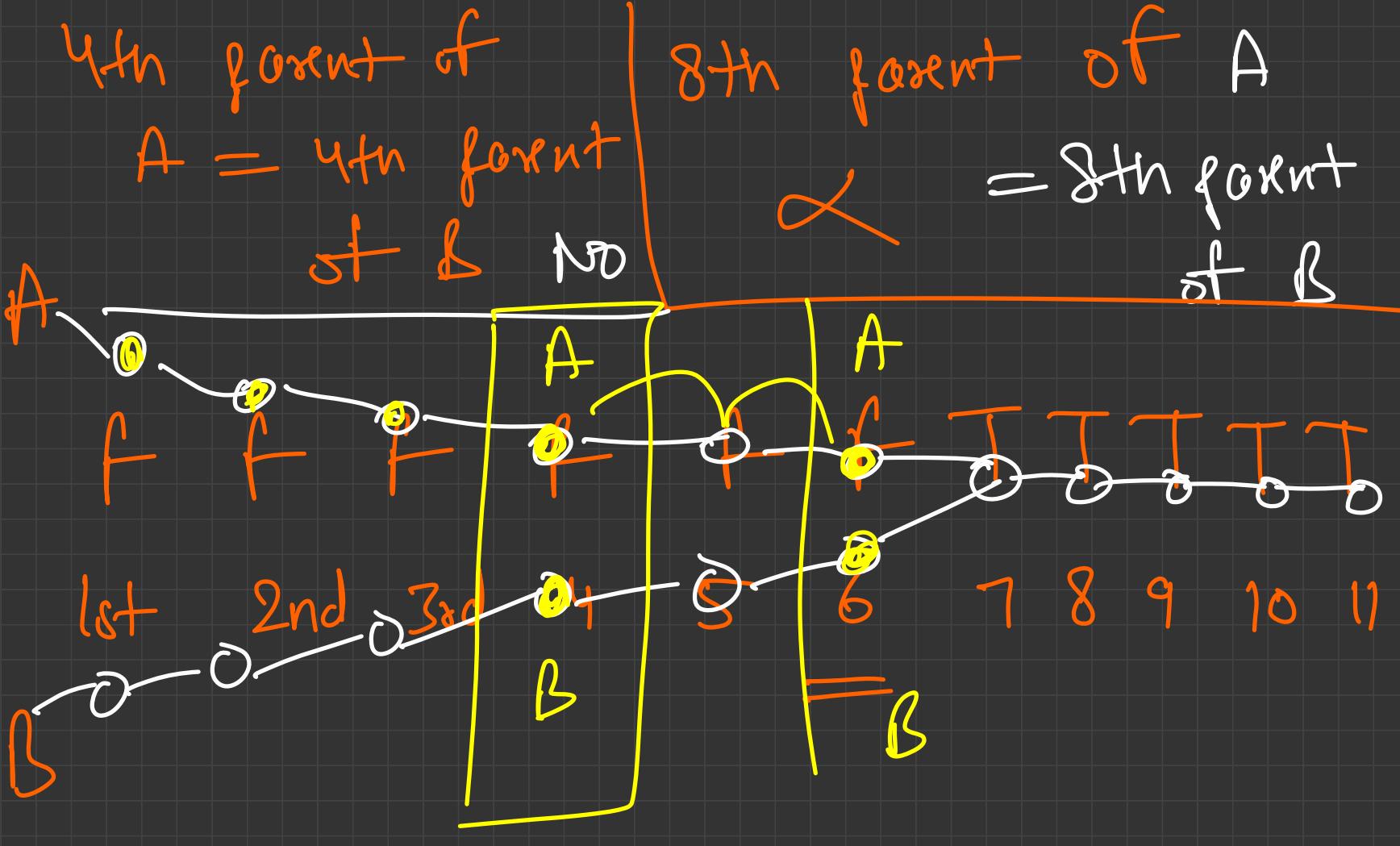
$O(\log n^2)$
 $\underline{\underline{(\log n \cdot \log n)}}$



① find last false and then
say that list front of A
= LCA

② find out the first True

= LCA



64 → O(1)

32 → O(1)

16 → O(1)

8 → O(1)

F F F F F F T T T T

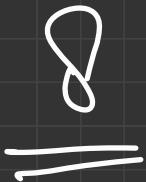
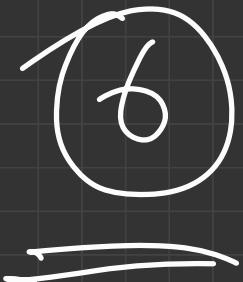
1 2 3 4 5 6 7 8 9 10 11

2^k th forent of any
node in O(1)

8 →

6th rosent is not
some

7th rosent is same



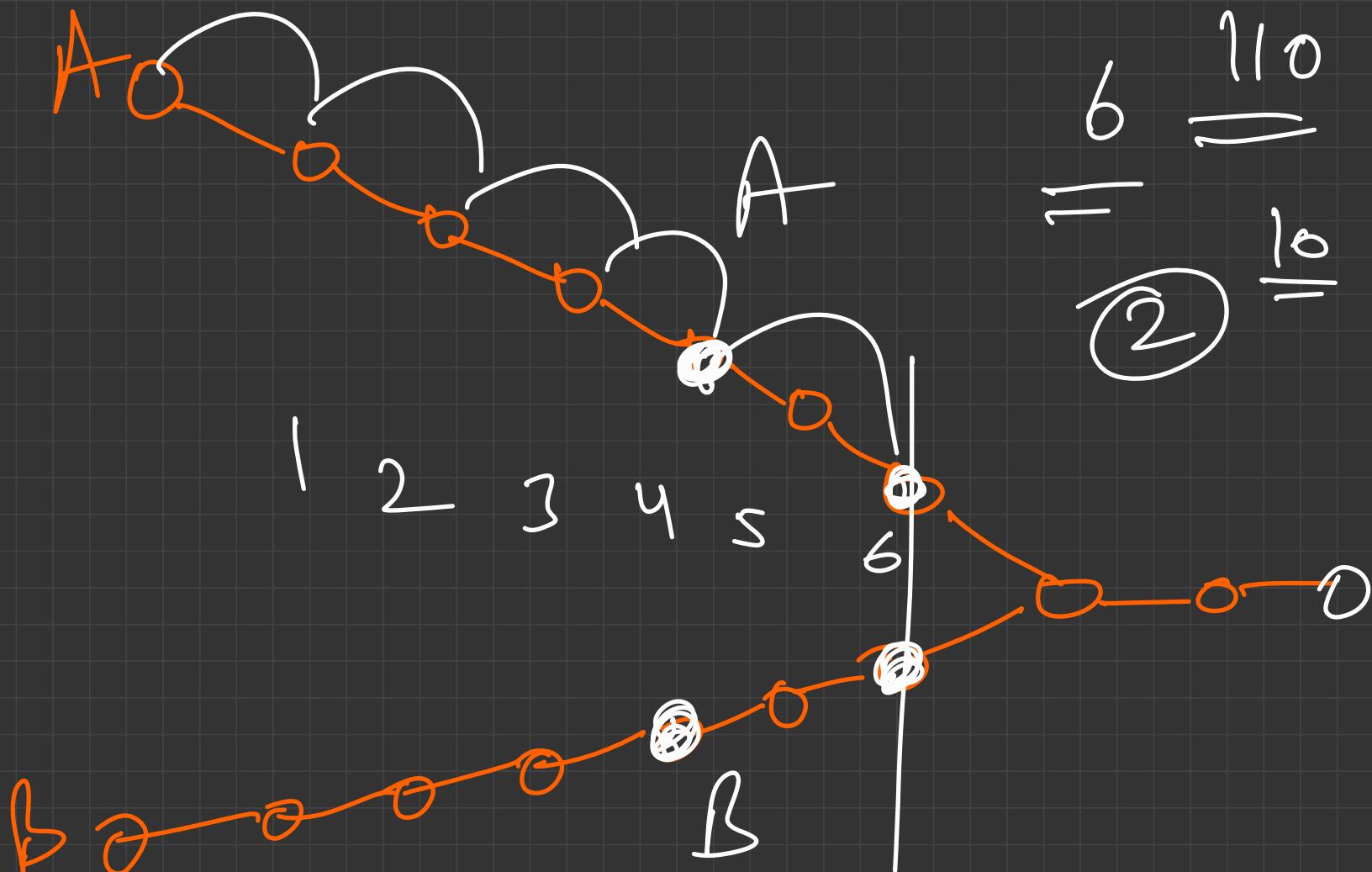
128 64 32 16 8 4 2 1

$$\sin(6) = \frac{1}{10}$$

~~1 0 0 0 0 0 0 1 0~~

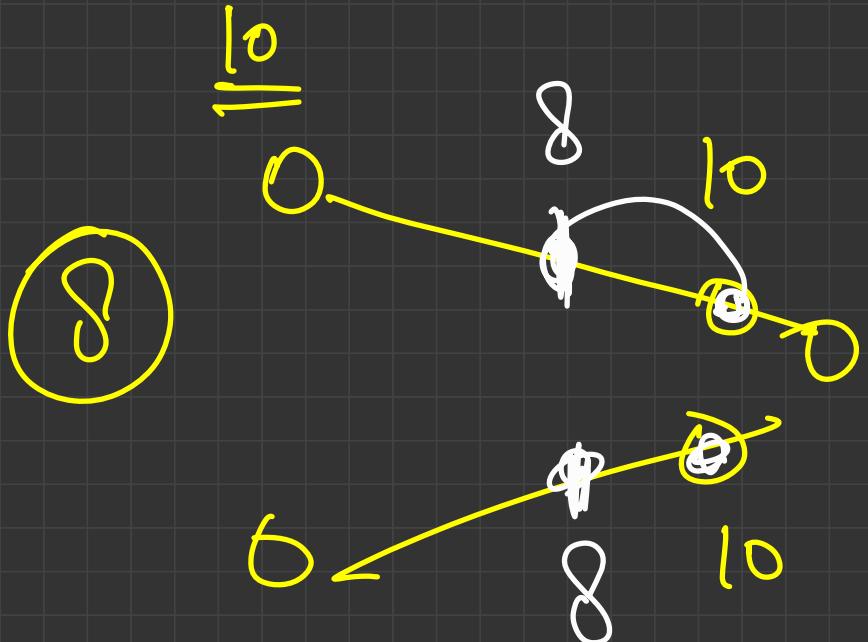
} 128th point of A == 128th of
points of B

O(1)

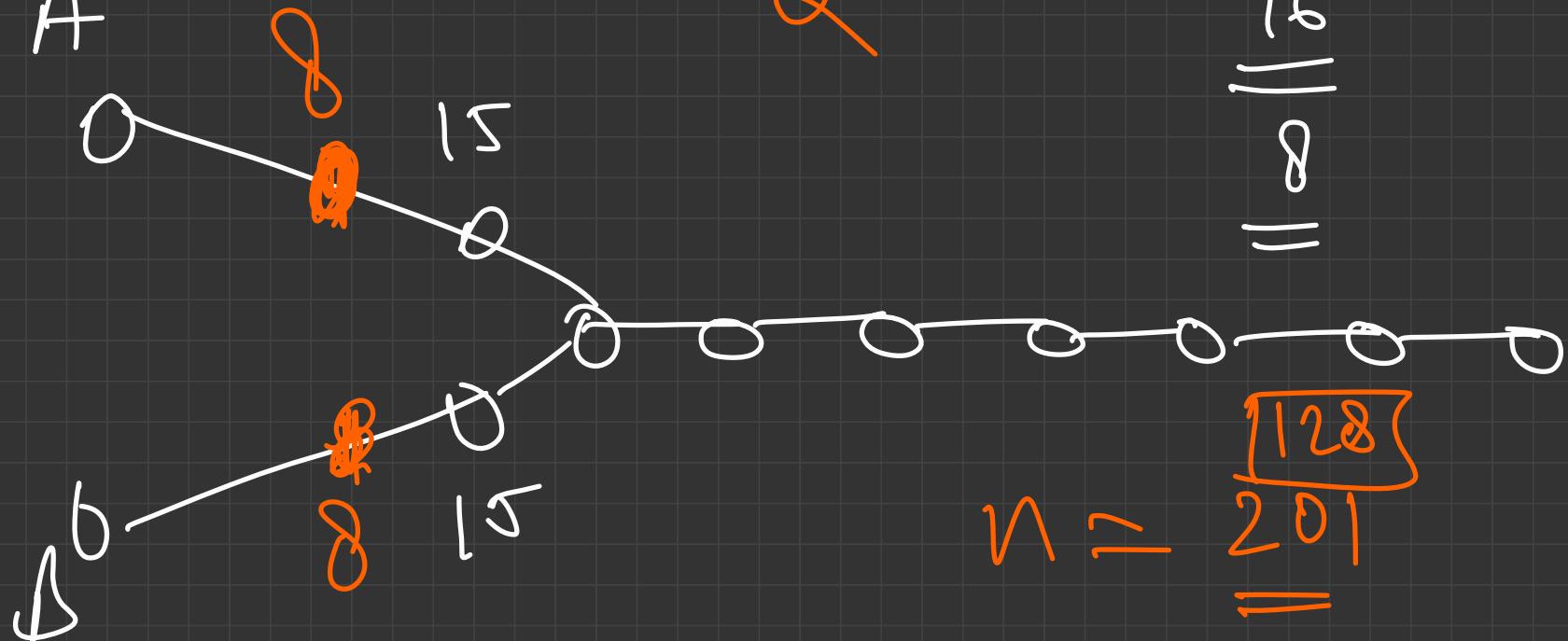


$$k = \frac{\alpha}{\sin(10^\circ)}$$

$$\begin{array}{rcl} \underline{\underline{64}} & \checkmark & \underline{\underline{8}} \\ \underline{\underline{32}} & \checkmark & \underline{\underline{4}} \\ \underline{\underline{16}} & \checkmark & \underline{\underline{2}} \end{array}$$



A



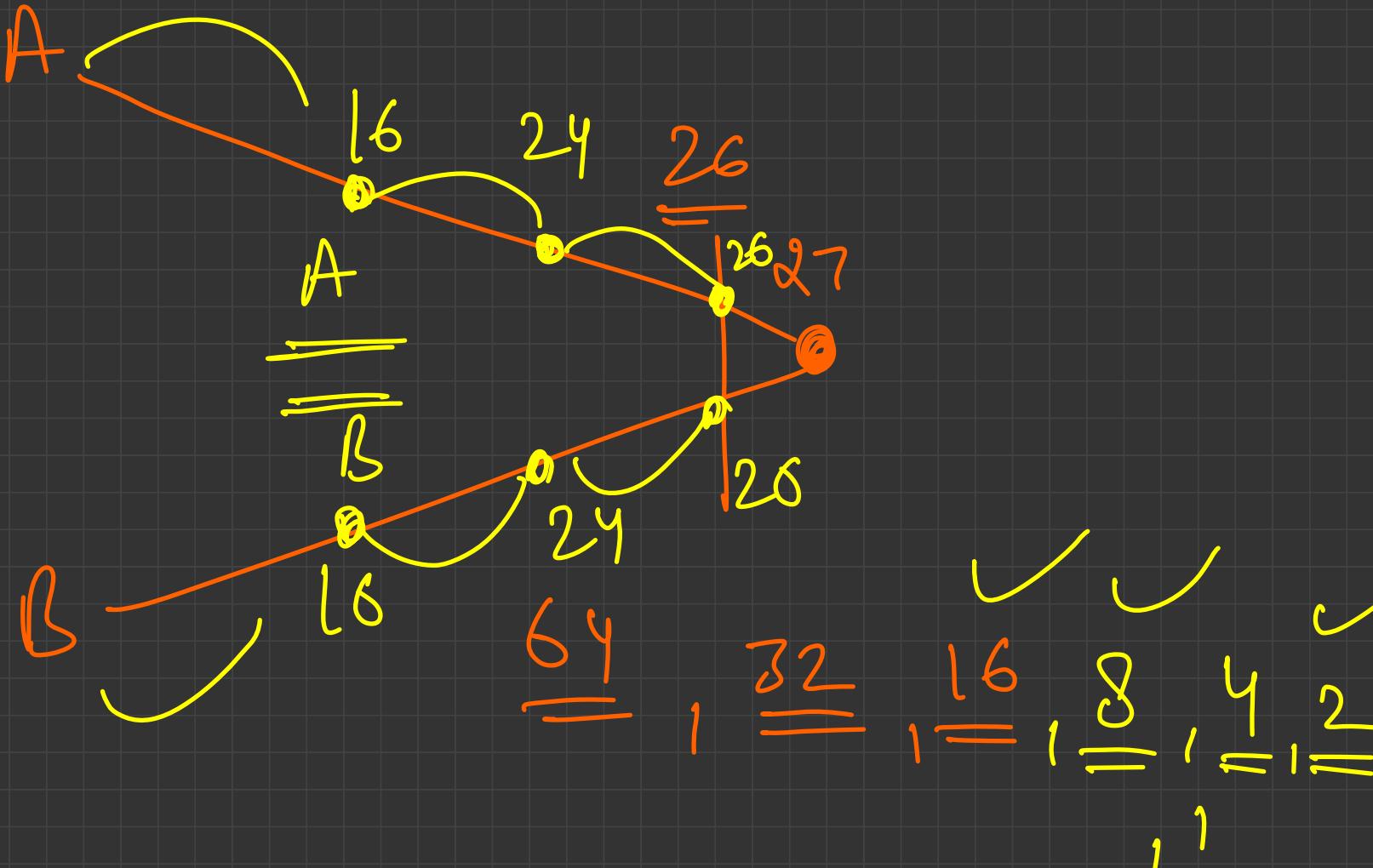
$$15 \\ \equiv$$

$$\log_2(n) \\ \equiv$$

$$32 \\ \equiv$$

$$16 \\ \equiv$$

$$n = 201 \\ \equiv$$



```
int lca( int x , int y ) {  
    if ( level(x) > level(y) )
```

O(log n) $\rightarrow x = \text{kth parent}(x, \text{level}(x) - \text{level}(y))$
O(log n) + if (level(y) > level(x))
 $y = \text{kth parent}(y, \text{level}(y) - \text{level}(x))$

both x and y at at same level

if (x == y) return "x";

```
for ( int k = log2(n) ; k >= 0 ; k-- ) {
```

+ $O(1)$ int faxnt_x = faxnt[k][x]

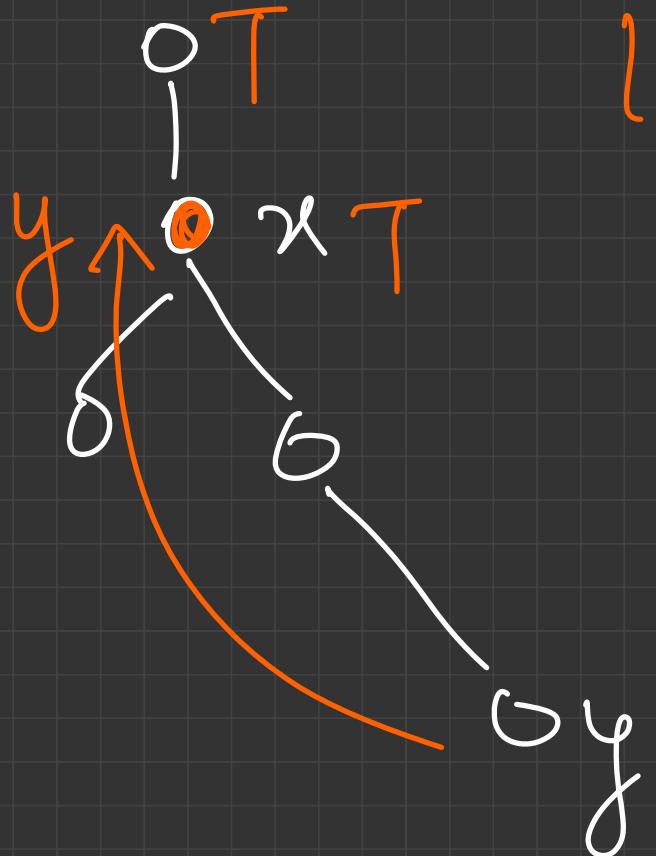
$O(1)$ int faxnt_y = faxnt[k][y]

$O(1)$ if (faxnt_x != faxnt_y) {

$x = faxnt_x$

$y = faxnt_y$

return faxnt[0][x] ;



$\text{lca}(x, y)$

$= n$ or y

,

- ① find the LCA fornt of every node $O(n)$
 ② find the level of every node $\overline{O(n)}$
 ③ binary lifting finding 2^k th front
 of every node
 $O(n \log n)$

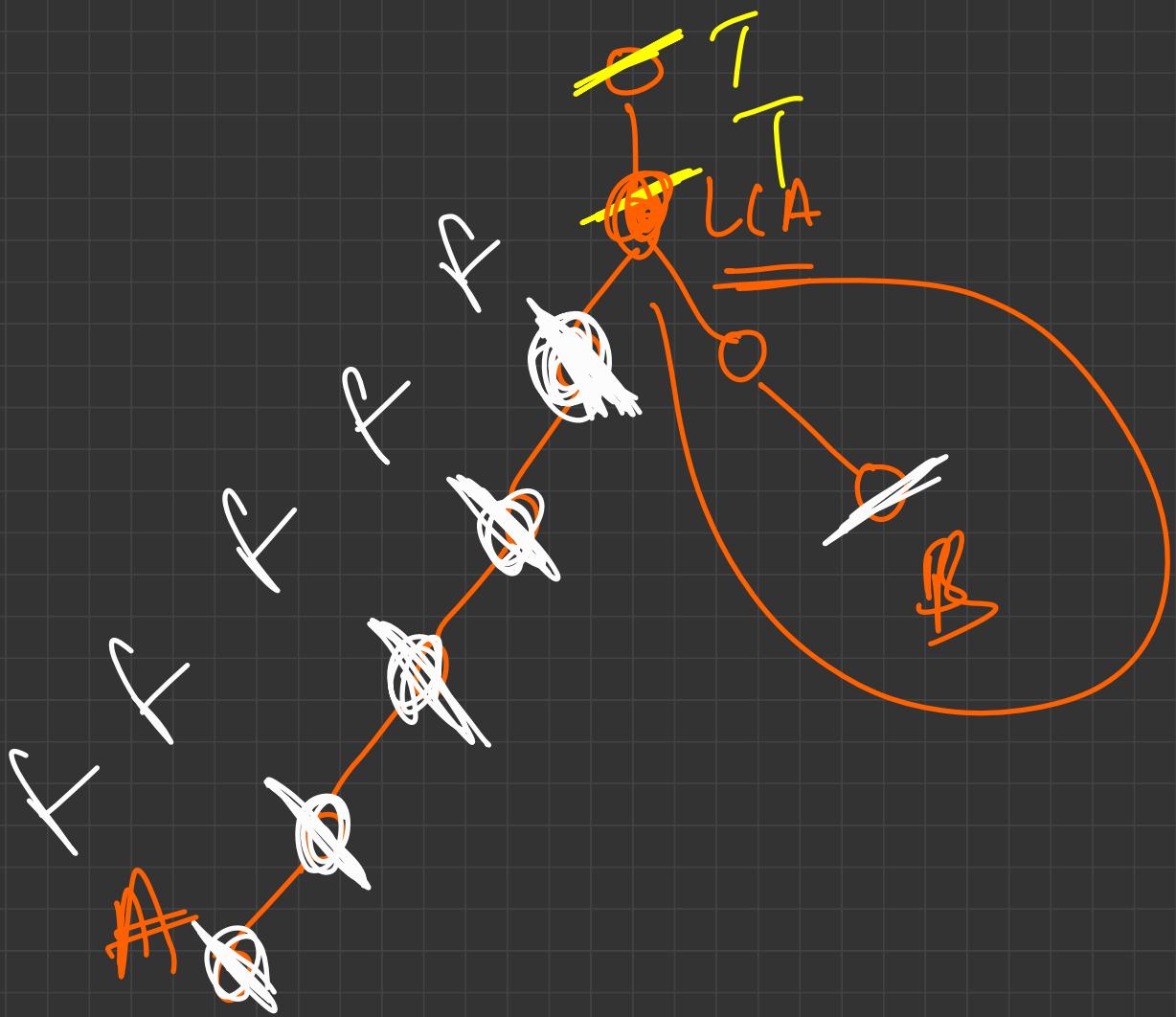
kth parent $\rightarrow O(k \log n)$

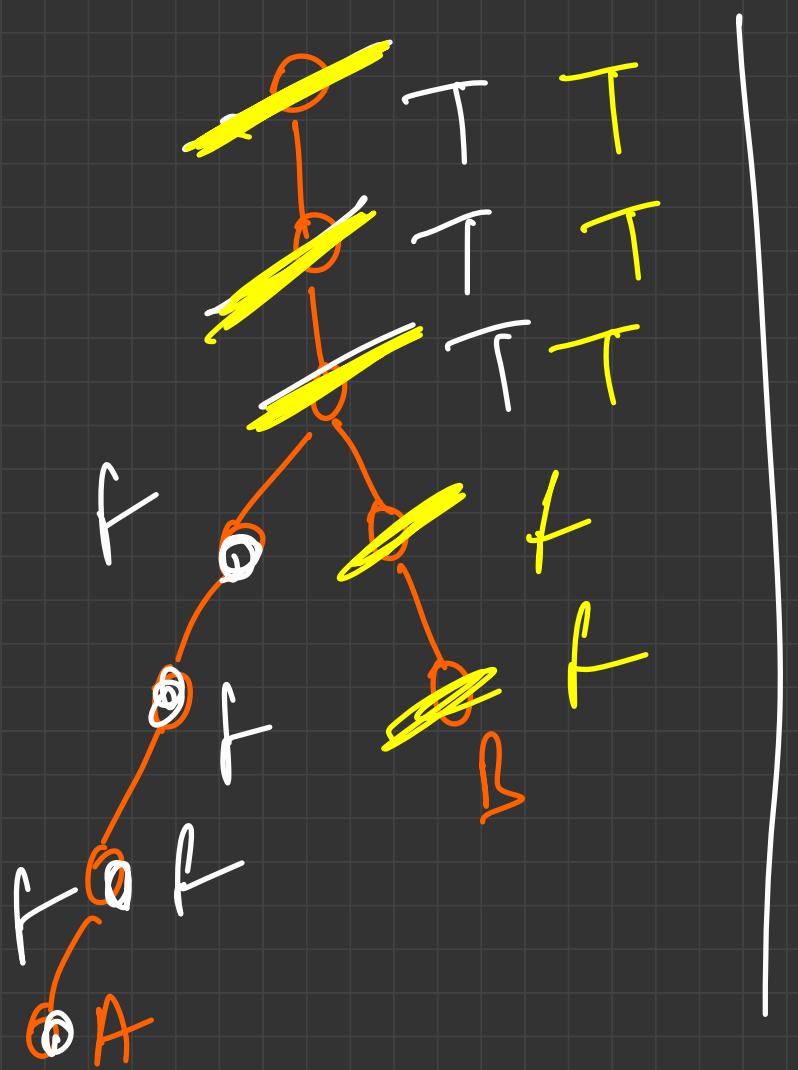
LCA $\rightarrow O(\log n)$

LCA

- Binary Searching $O(\log n * \log n)$
- Avoiding binary search $O(\log n)$
- More elegant solution using in-time and out-time $O(\log n)$







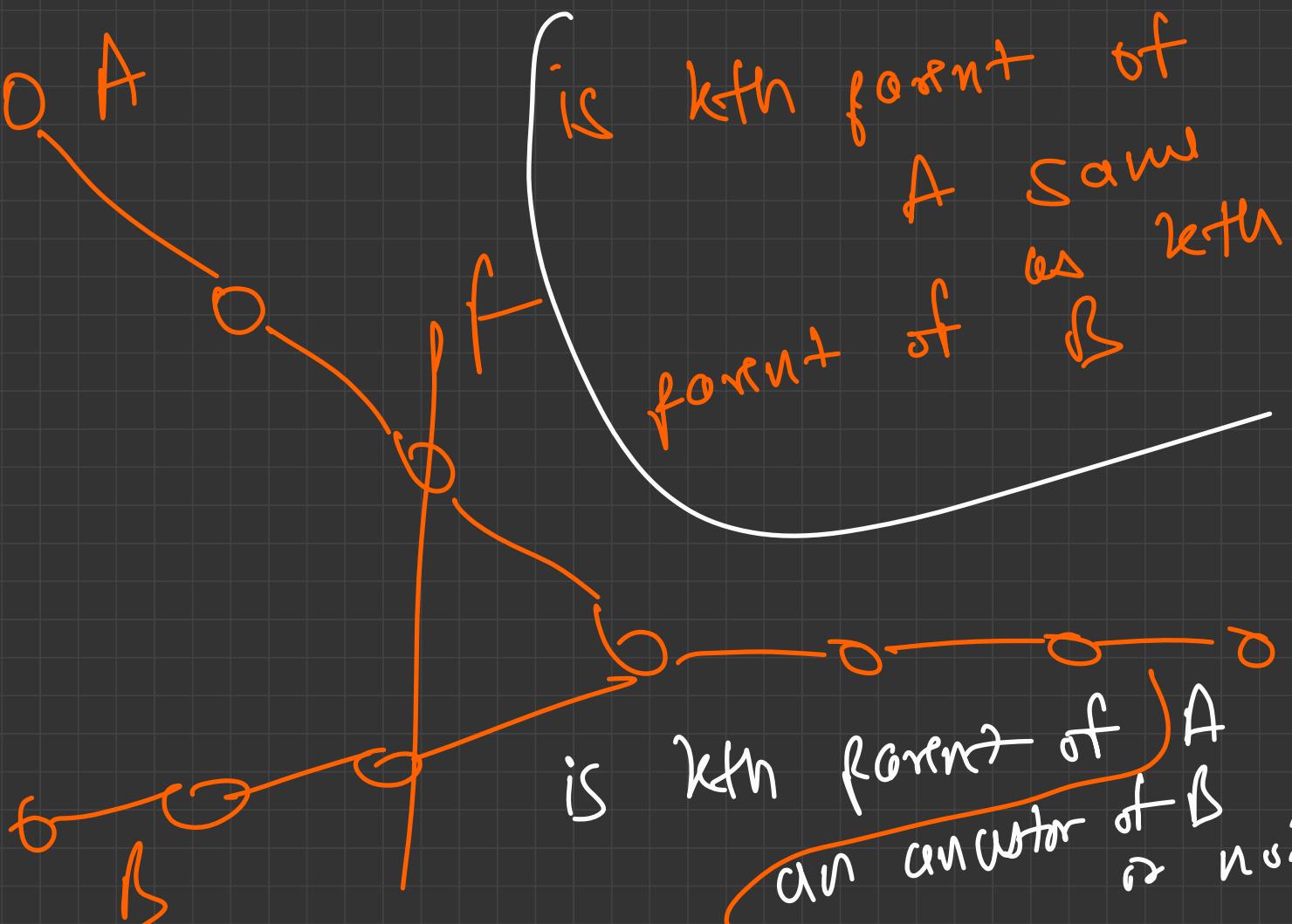
① find out the last parent of A which is not an ancestor of B

② find out the last parent of B which is not an ancestor of B

① is A an ancestor of B
return A

② is B an ancestor of A
return B

③ find the last parent of A which
is not an ancestor of B and
return the last parent or else null



```
int LCA (int n, int y)
if ( is-ancestor (n, y) )
    return n
if ( is_ancestor (y, n) )
    return y
for (int k = log2(n); k >= 0; k--) {
    int parent_n = parent[k]/n
    if ( ! is_ancestor (parent_n, y) )
```

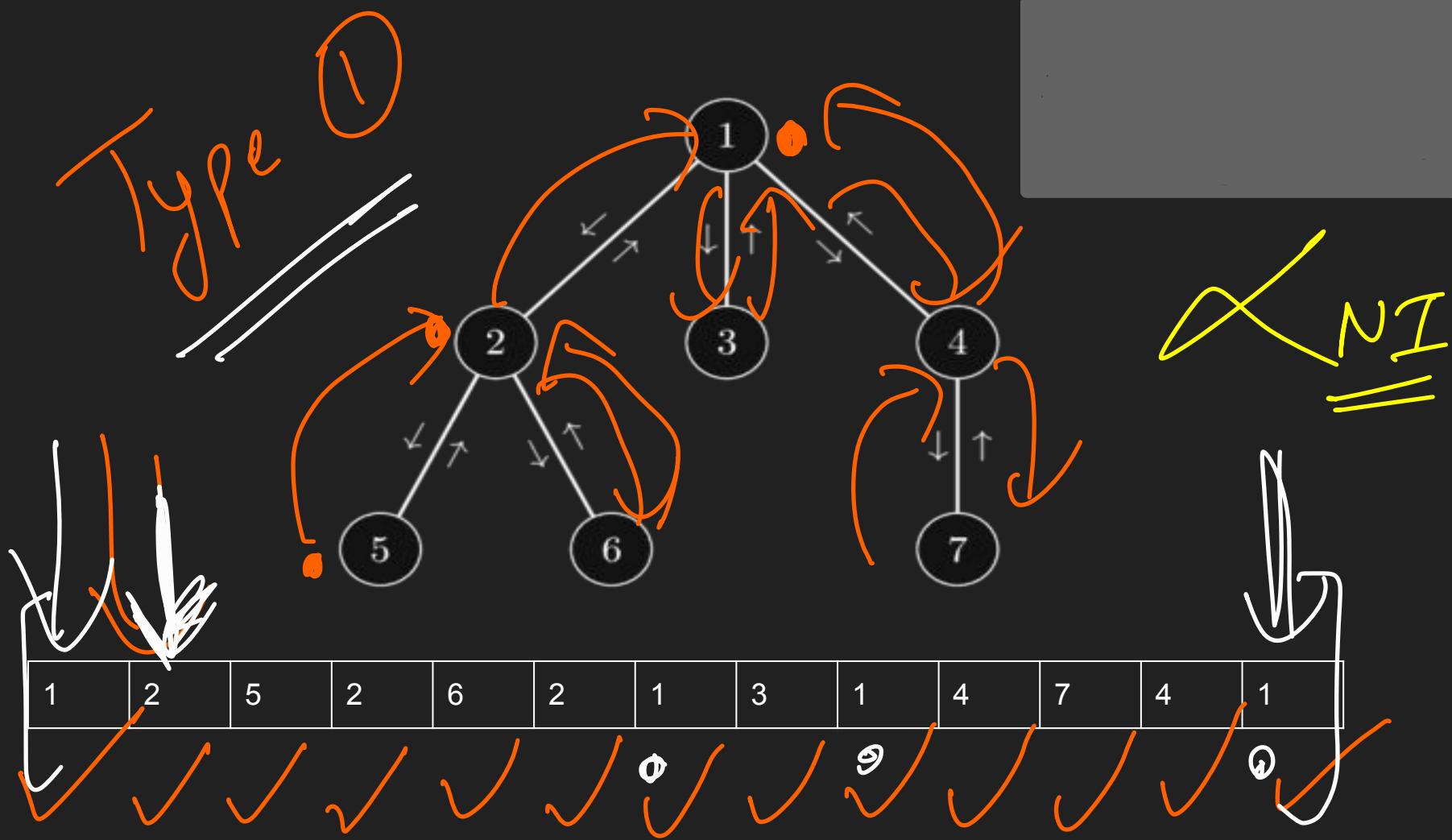
$\lambda = \text{parent}_\sim x$

3

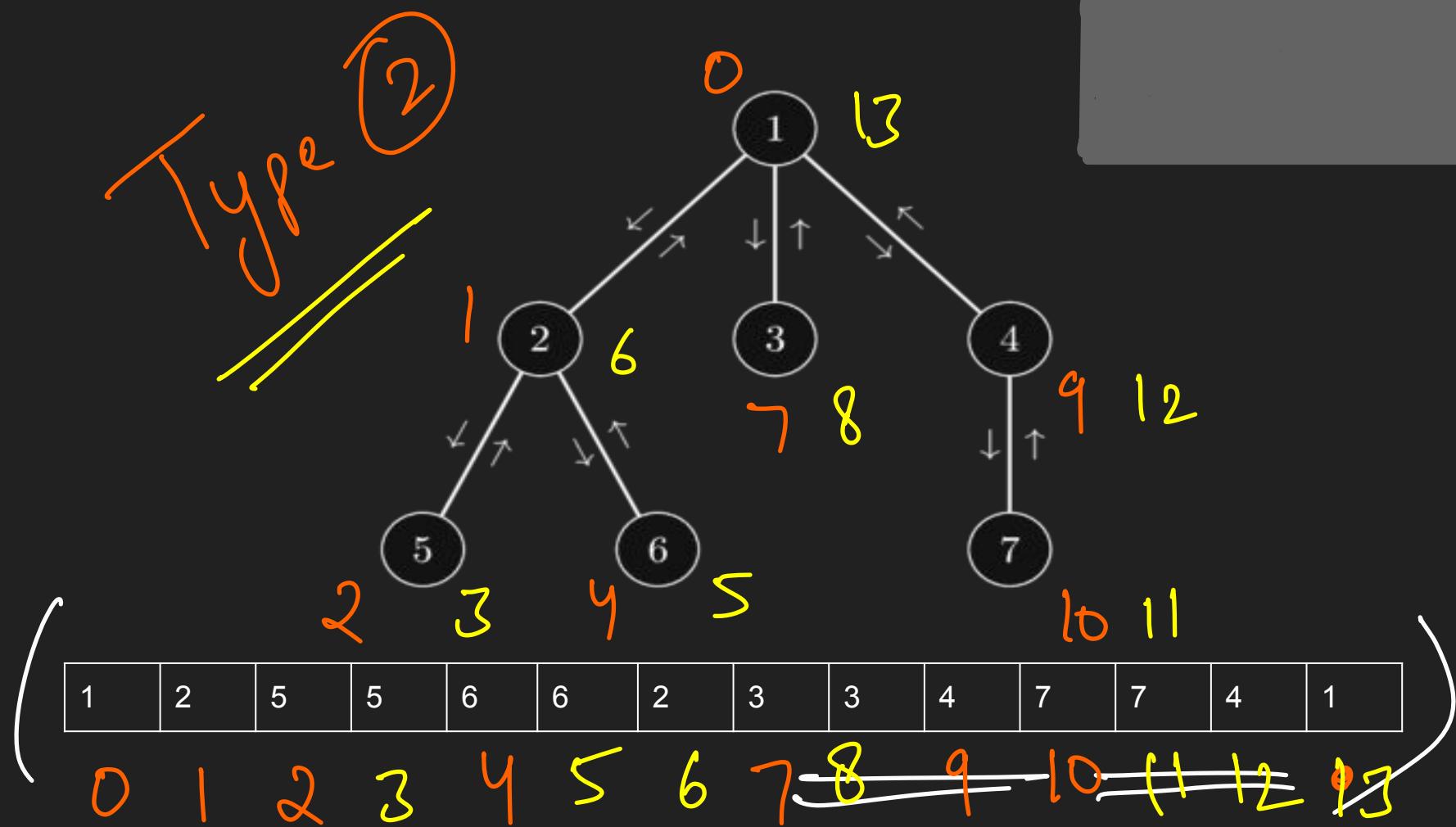
return parent($\sigma[x]$)

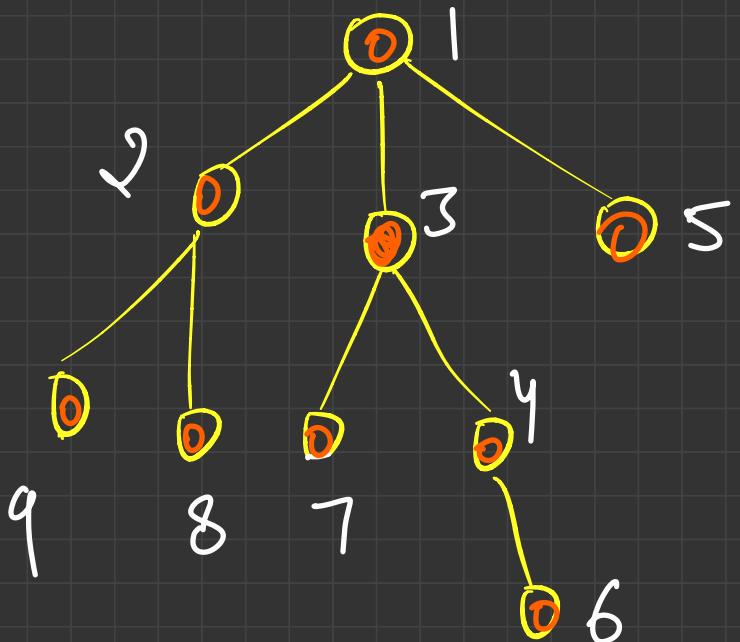
Euler Tour

- In-time Out-time relation
- Time complexity
- ✓ Code
- Problem - Queries (Add X to subtree)



Type ②



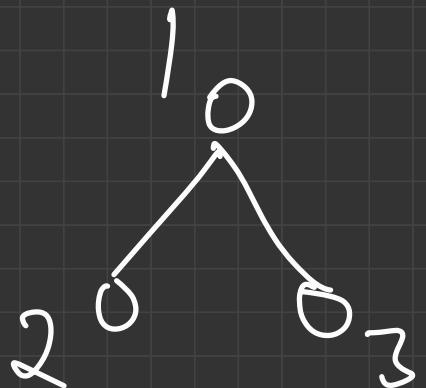
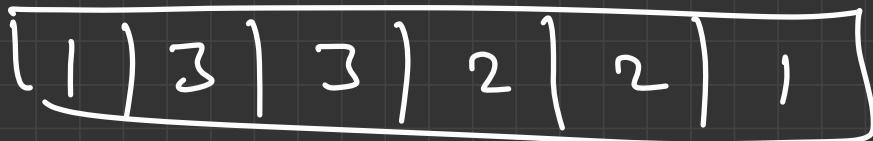


+5 to all nodes
in subtree
of 3



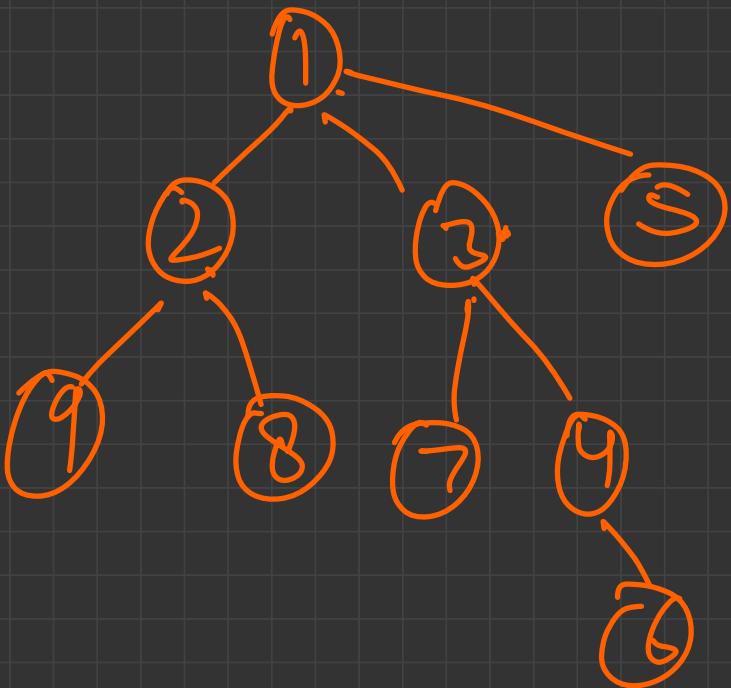
1 2 9 9 8 8 2 3 7 7 4 6 6 4 3 5 5 1

Euler tour

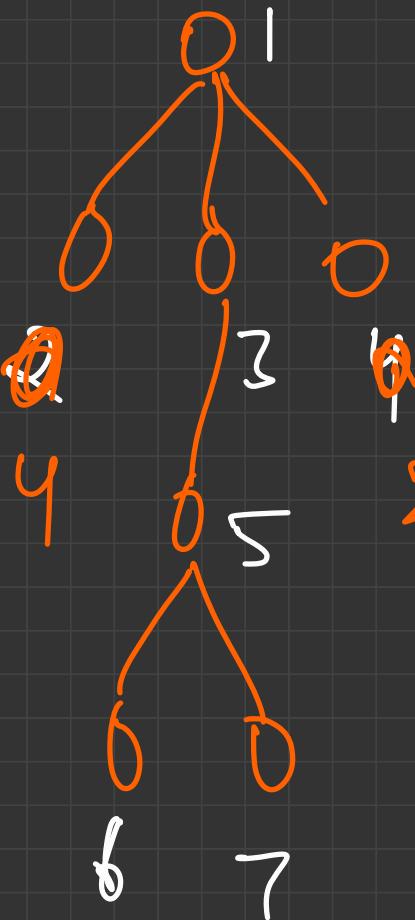


1 2 9 9 8 8 2 3 7 7 4 6 6 4 3 5 5 1

0 0 0 0 6 6 0 3 - 0 0 , 0 0 0 0 0



$O(n)$

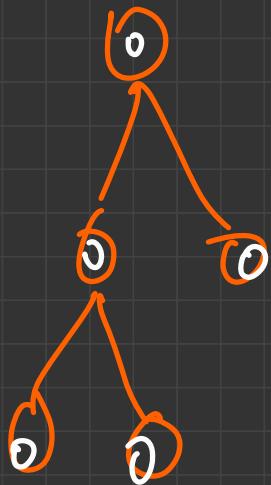


| Tree → multiple euler tours

| euler tour → | Tree

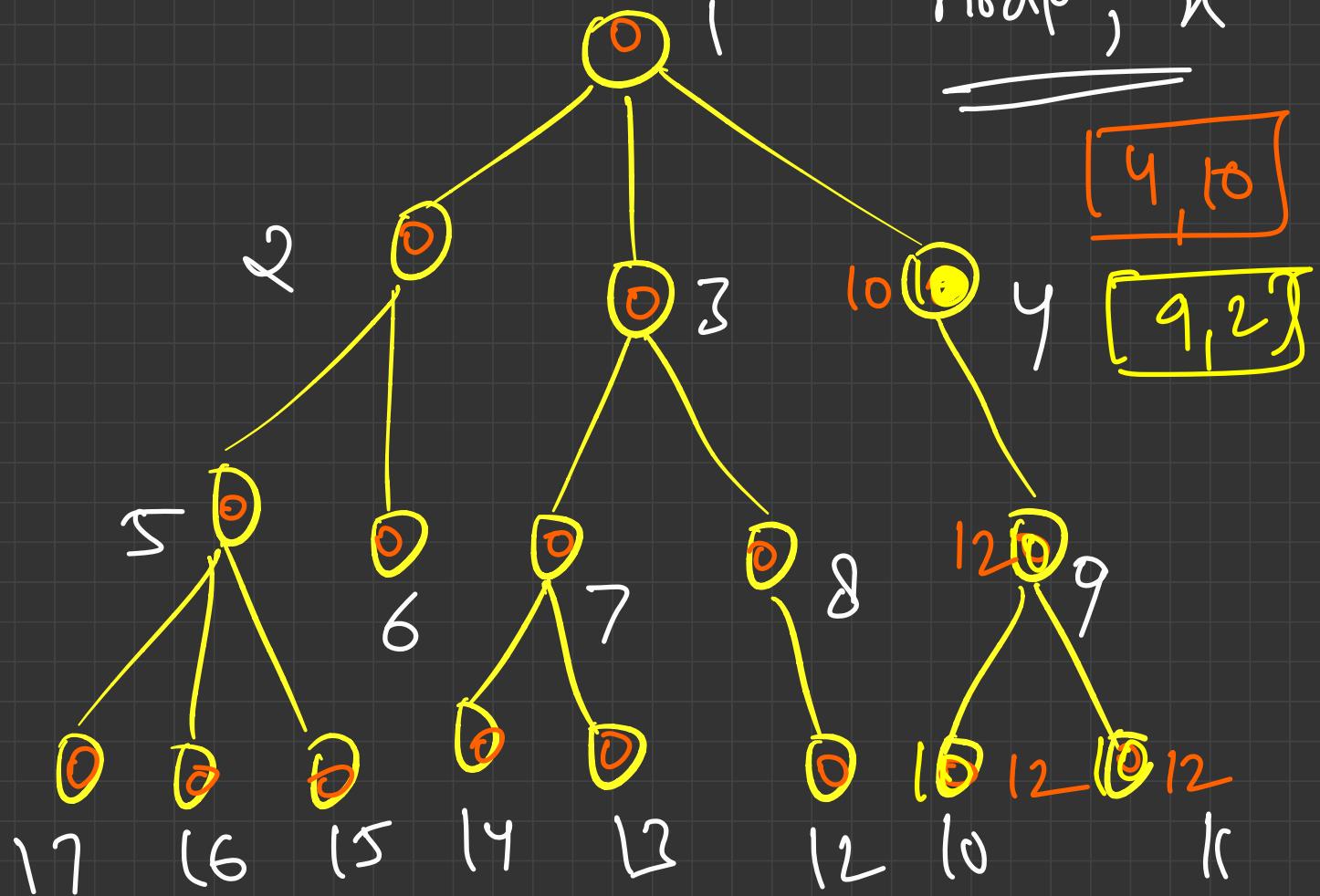
1 2 2 3 5 6 6 7 7 5 3 4 4 1
| 4 4 2 2 3 5 6 6 7 7 5 3 1

① Given a tree with n nodes perform the following queries and return the final tree all values are initially filled with 0



1 $\&$ Q queries

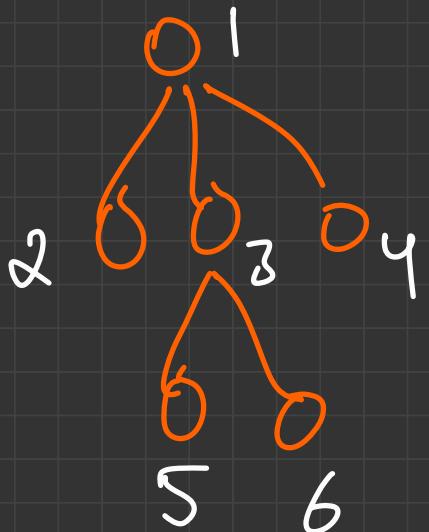
$$N \leq 10^5, Q \leq 10^5$$



node⁰, n

$[4, 10]$

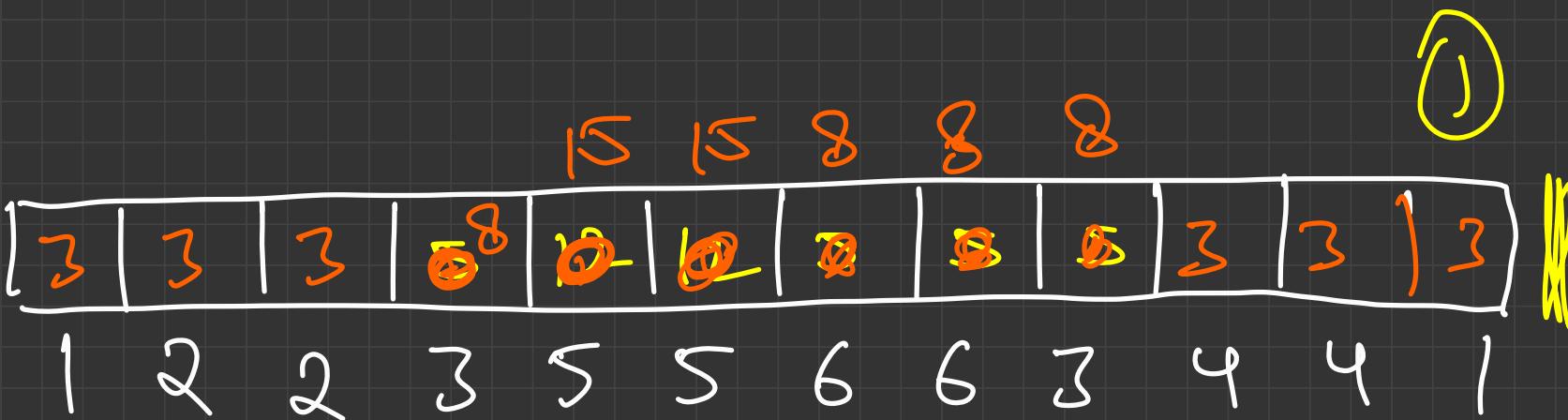
$[9, 12]$



Add 5 for subtree(3)

Add 7 for subtree(5)

Add 3 for subtree(1)



3												2
0	0	5	7	0	-7	0	0	-5	0	0	0	1
1	2	2	3	5	5	6	6	3	4	4	1)

Add 5 for subtree (3)

add γ
from $(\emptyset, 0)$

Add 7 for subtree (5)

$$\text{add}(l) + \gamma$$

Add 3 for subtree (1)

$$\text{add}(d+1) = n$$

0	0	0	0	0	0	0
0	1	2	3	4	5	6

Add 5 to
 all values
 from
 $\underline{\underline{(1,3)}}$

0	5	5	5	0	0	0
0	1	2	3	4	5	6

Add 4 to
 all values
 from $\underline{\underline{(2,5)}}$

0	5	9	9	4	4	0
6	1	2	3	4	5	6

$$\boxed{[5|4| \quad | -5 | \quad | -4]}$$

$$0 \quad 1 \ 2 \quad] \ 4 \leq 6$$

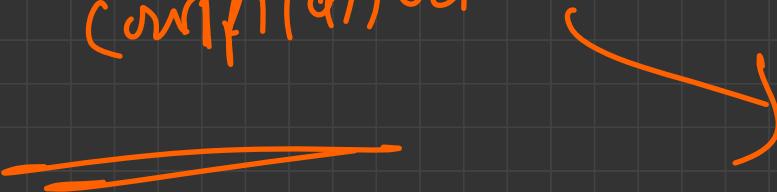
$$0 \ 5 \ 9 \ 9 \ 4 \ 4 \ 0$$

$$\cos(\theta) \pm \frac{\pi}{2}$$
$$\cos(\theta + 1) = x$$

$O(n)$ time to generate euler tour

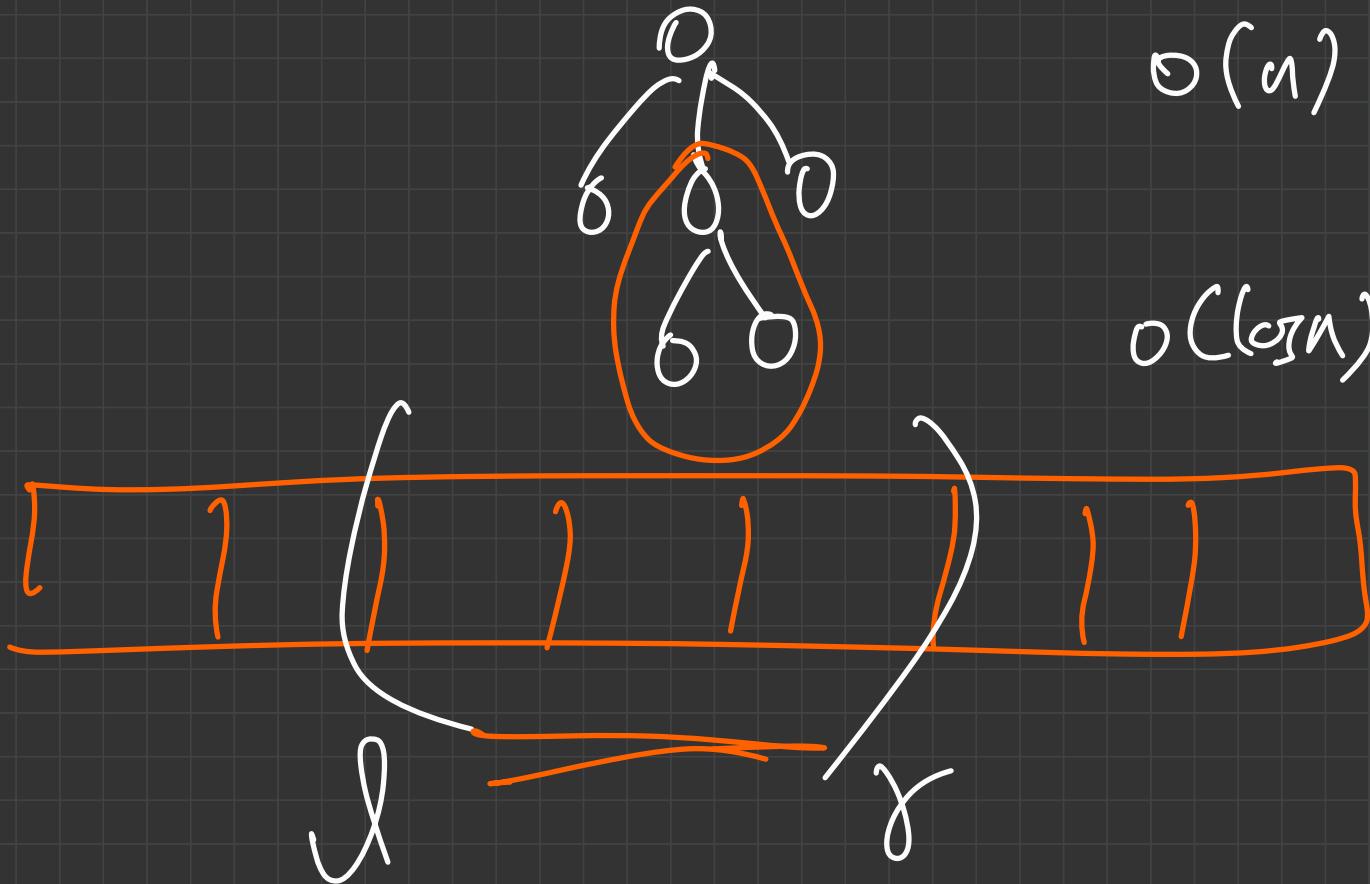
$O(1)$ for every query

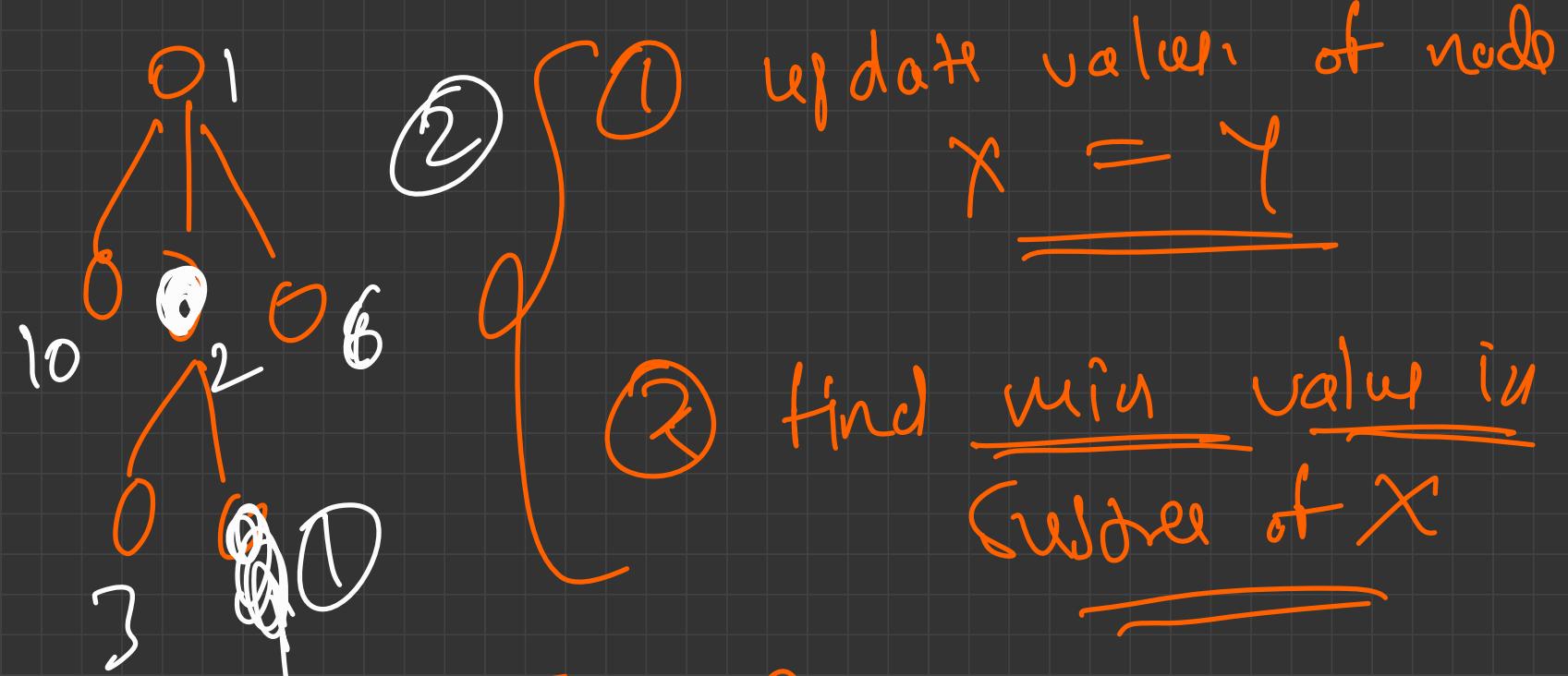
$O(n)$ for configuration



$O(n)$ brute
force

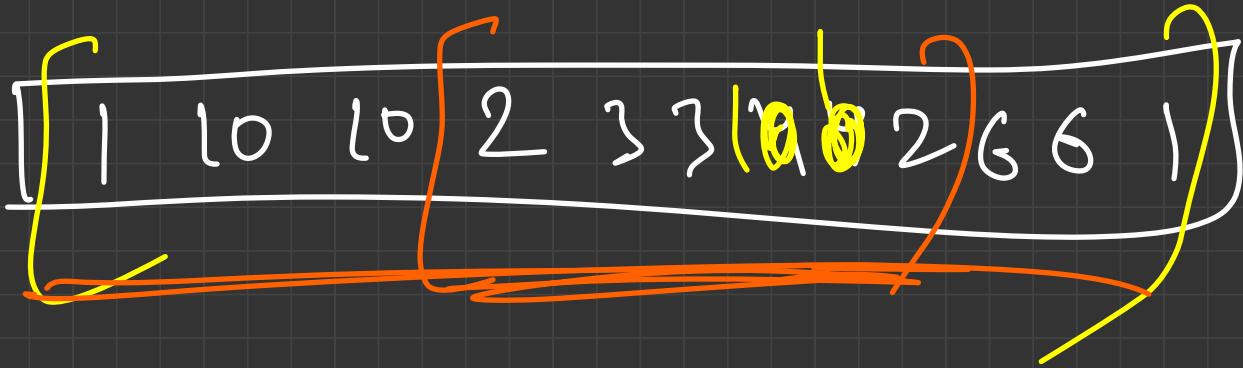
$O(\log n)$ segm.
tree



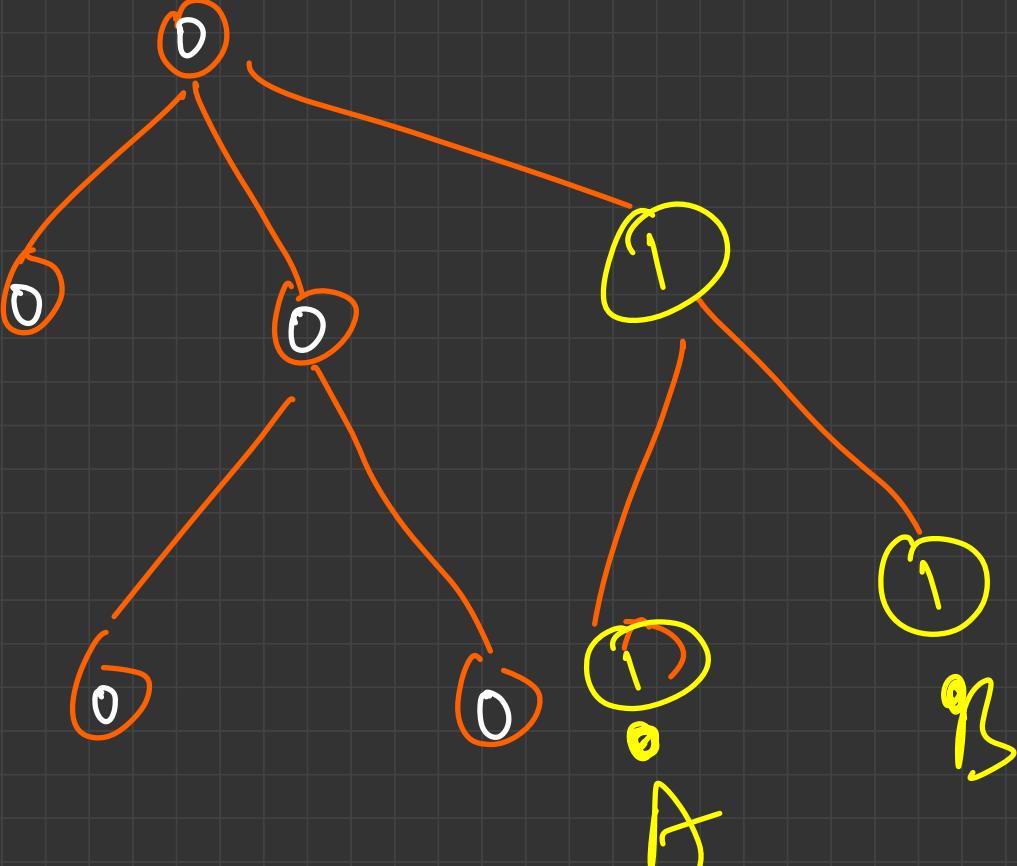


$$N \leq 10^5$$

$$Q \leq 10^5$$



Segment Tree



A

B



A



lga



A'



B



B

A



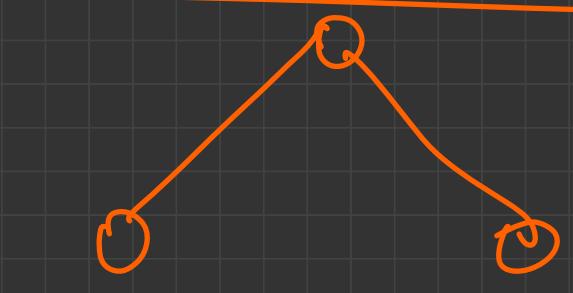
A, B

\equiv

if $\text{level}(A) > \text{level}(B)$

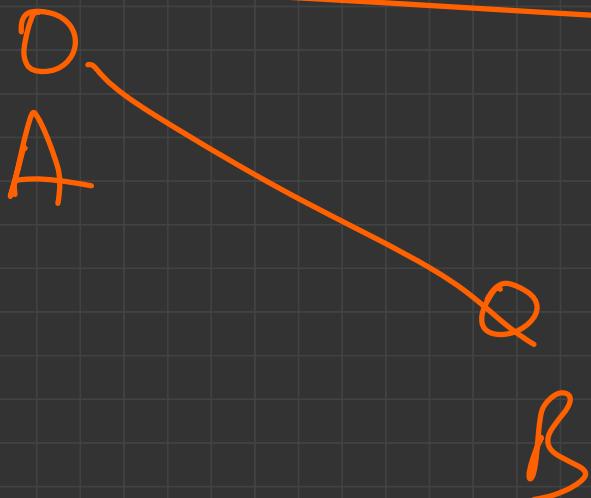
$\text{swap}(A, B)$

lca

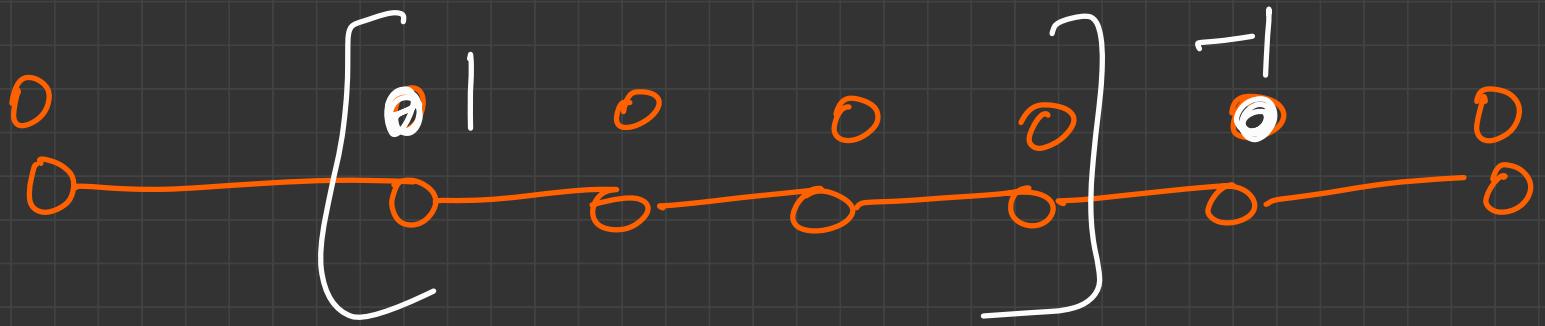


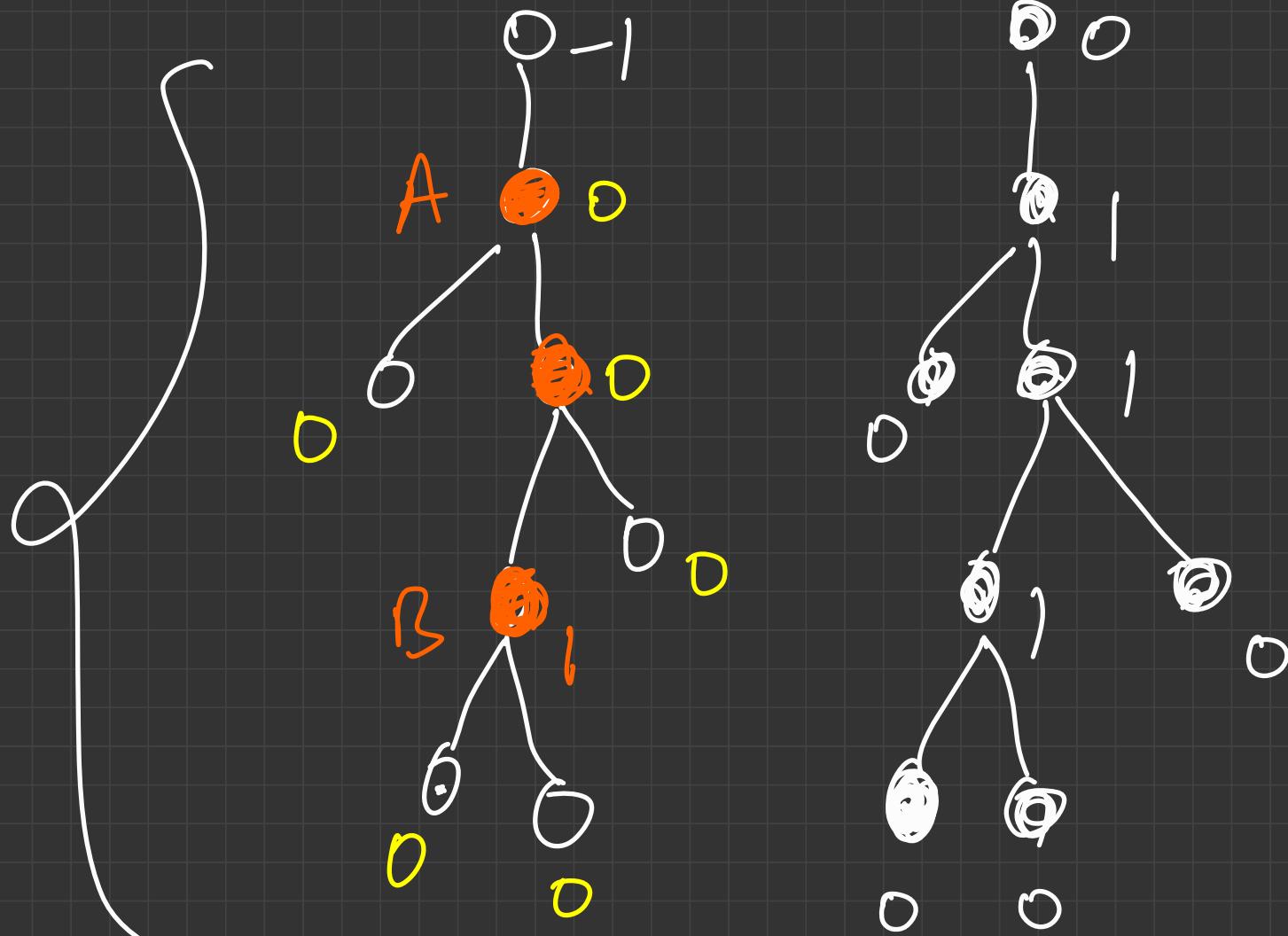
A

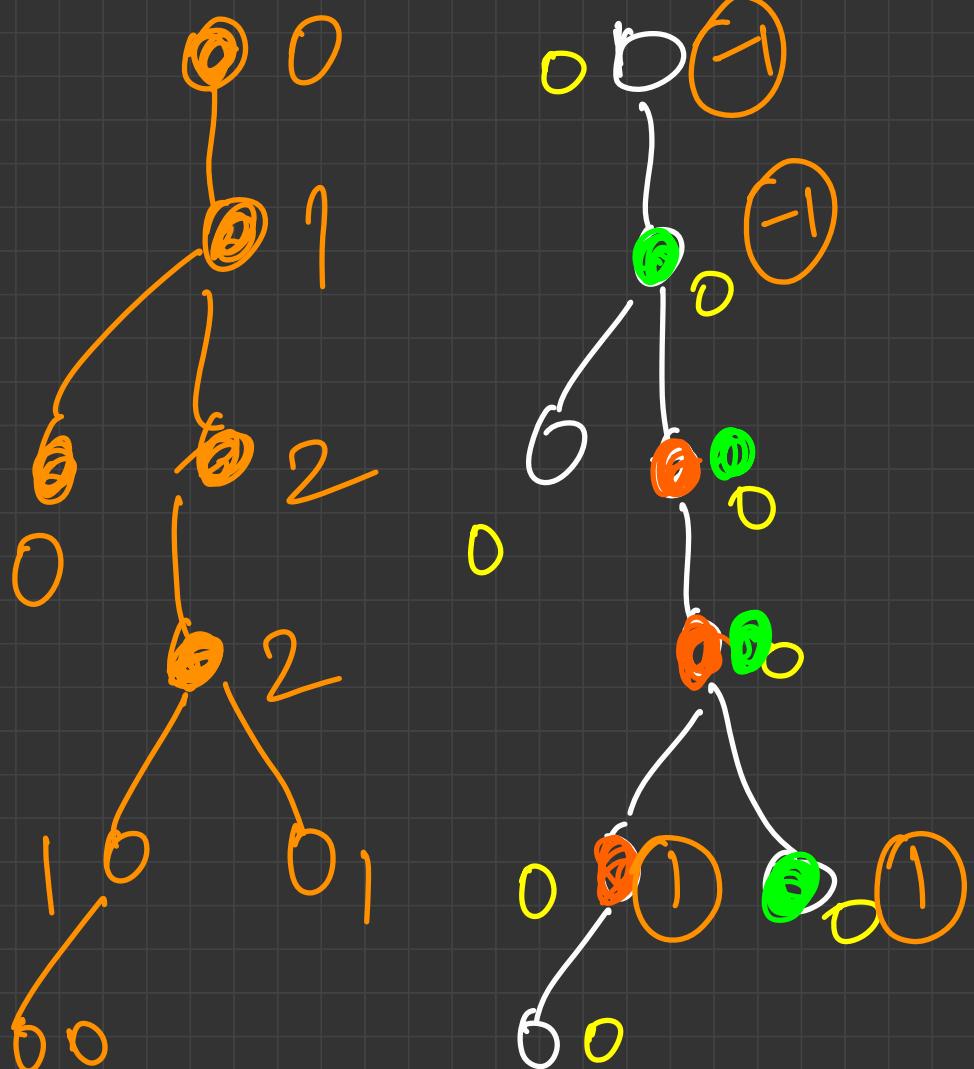
B



B

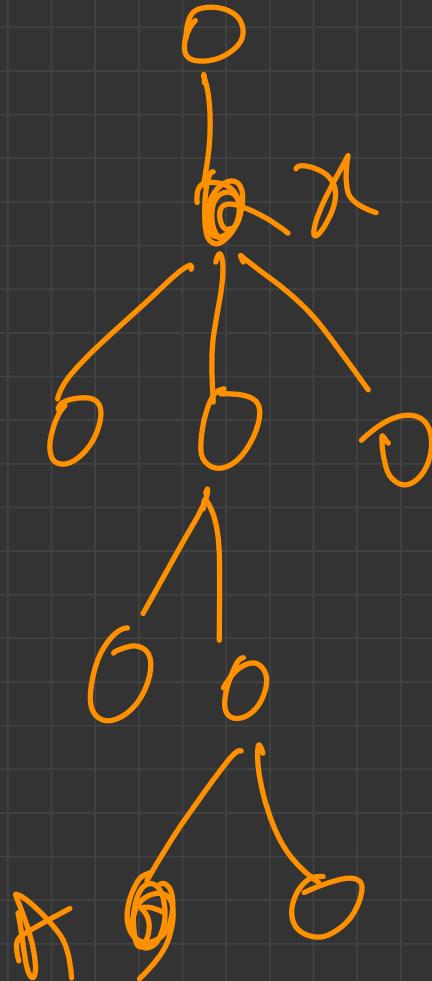






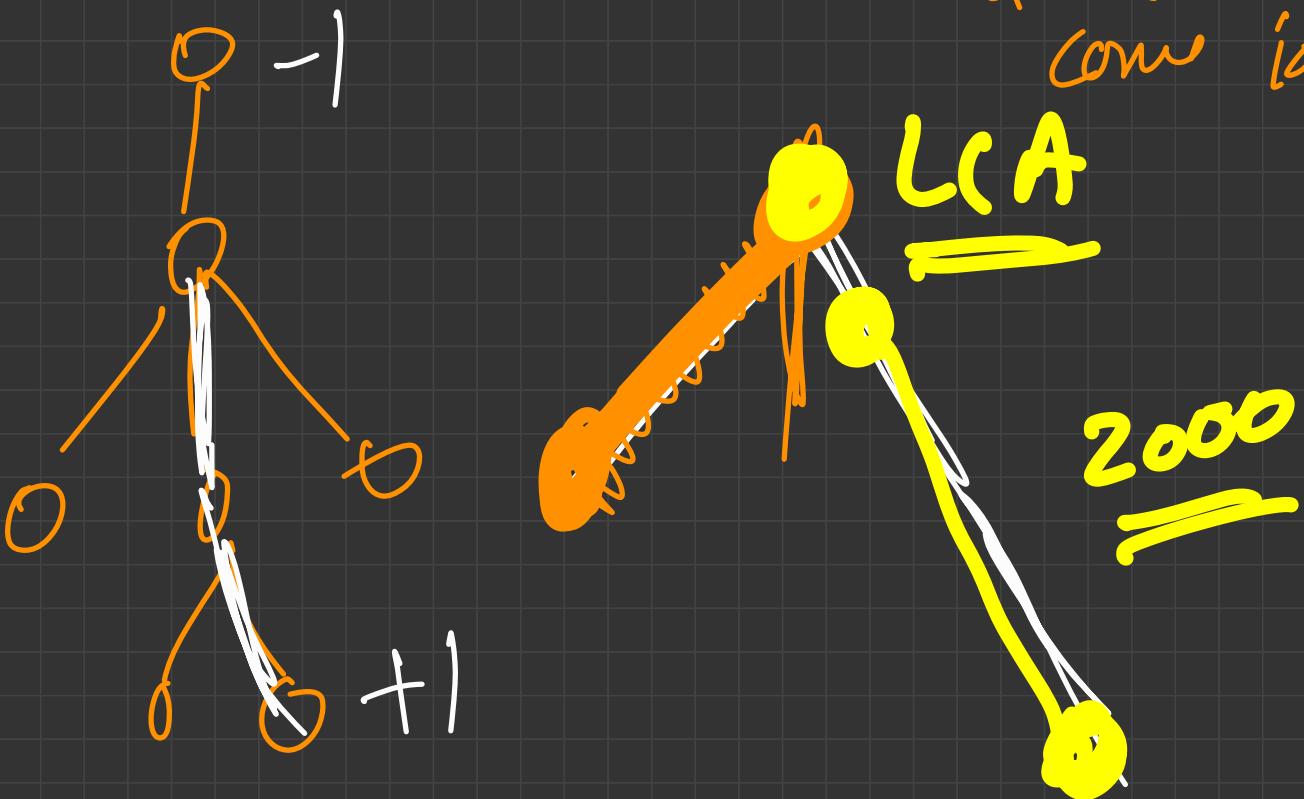
Add 1 to all green nodes

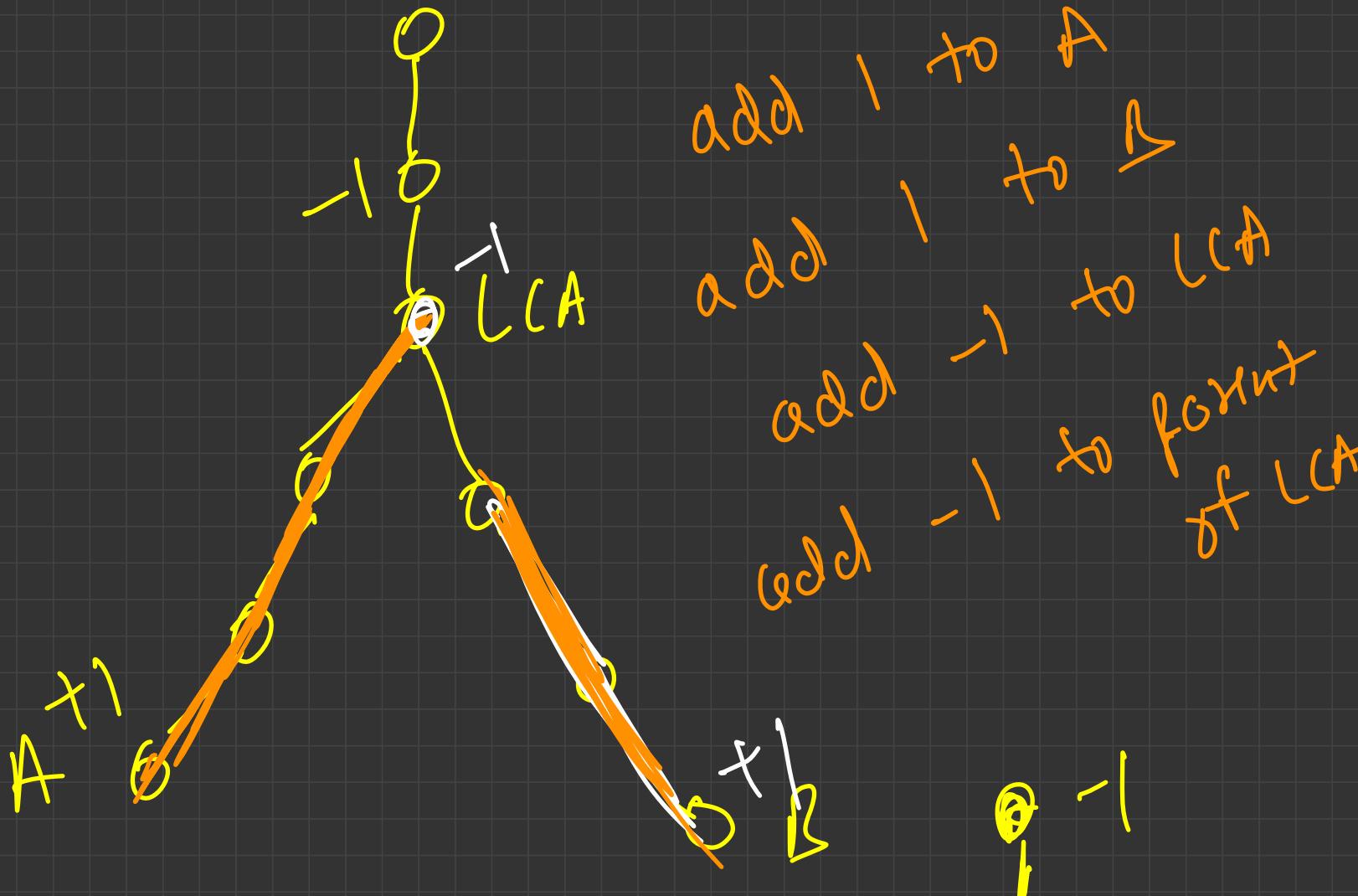
Add 1 to all orange nodes

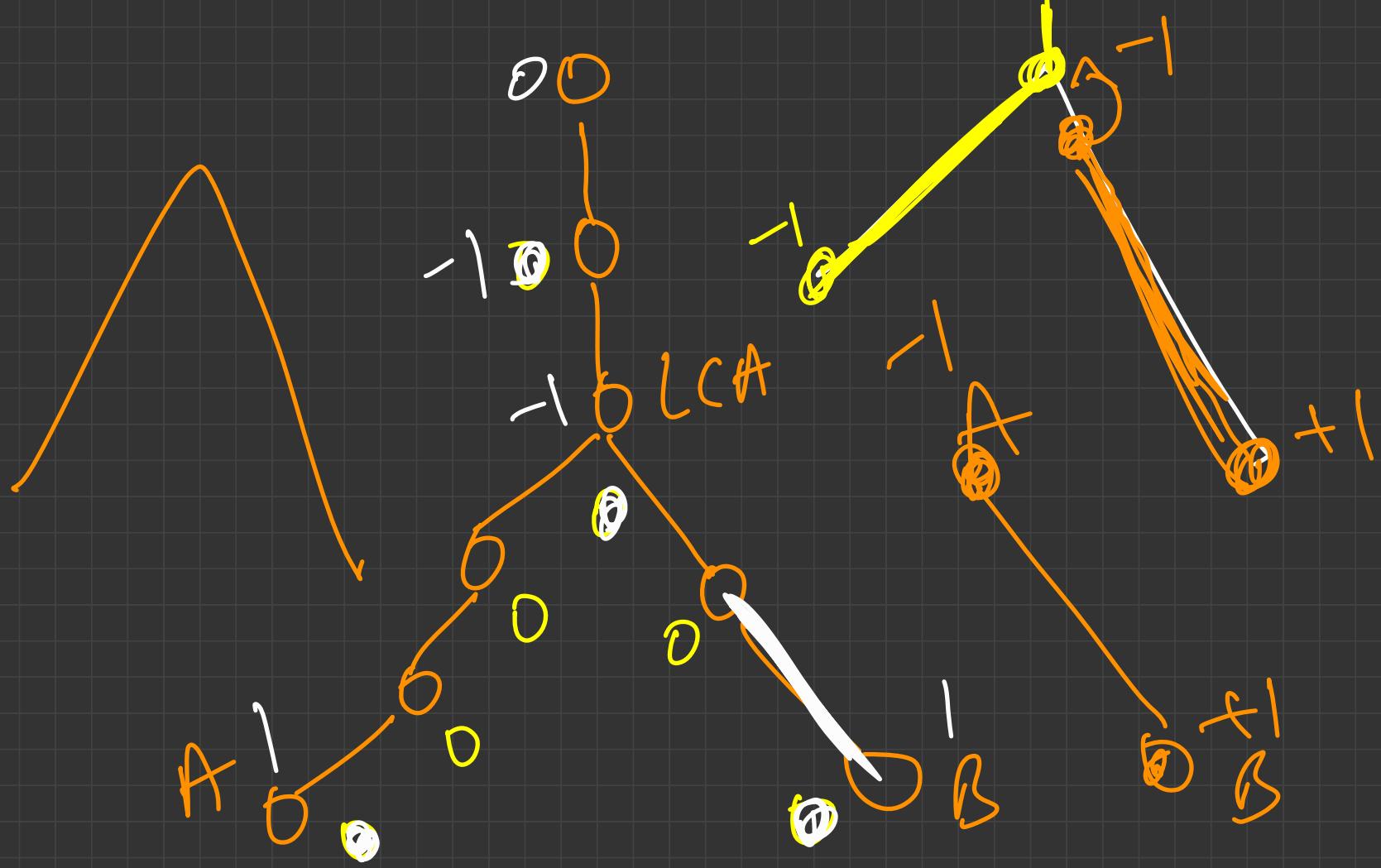


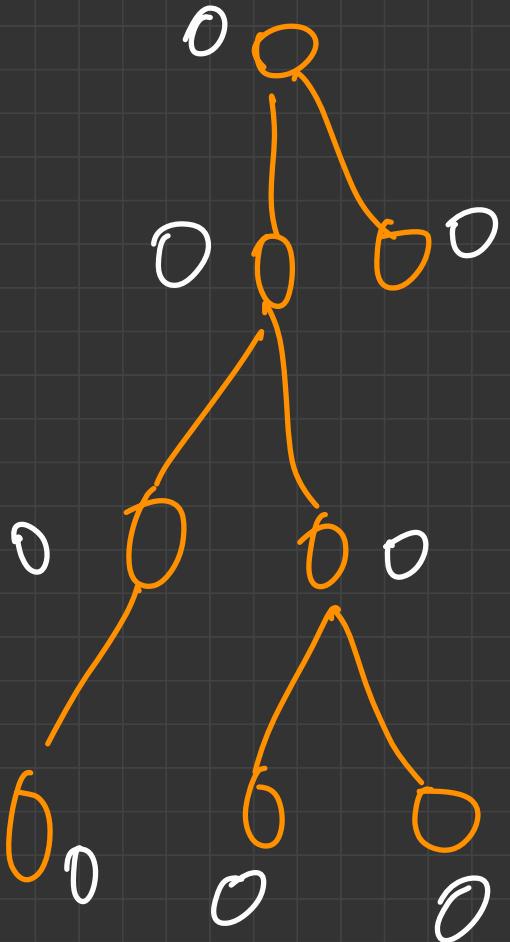
Add | to
all the
nodes
and
from
A to X

find how many factors does
a node
have in







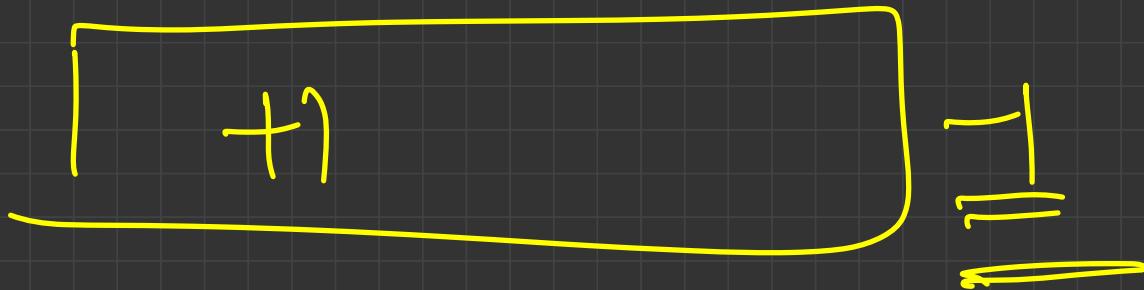


if A is an ancestor of
 B
+ 1 cut \underline{B}
- 1 cut Parent of A

① if $\text{level}(A) > \text{level}(B)$ $O(1)$
 $\text{swap}(A, B)$ $\underline{\underline{O(1)}}$

② if A is an ancestor of B $O(1)$
+1 at B, -1 at left

③ +1 at A
+1 at B $O(\underline{\log n})$
-1 at L(A), -1 at left front of L(A)



Lat differeny
andy