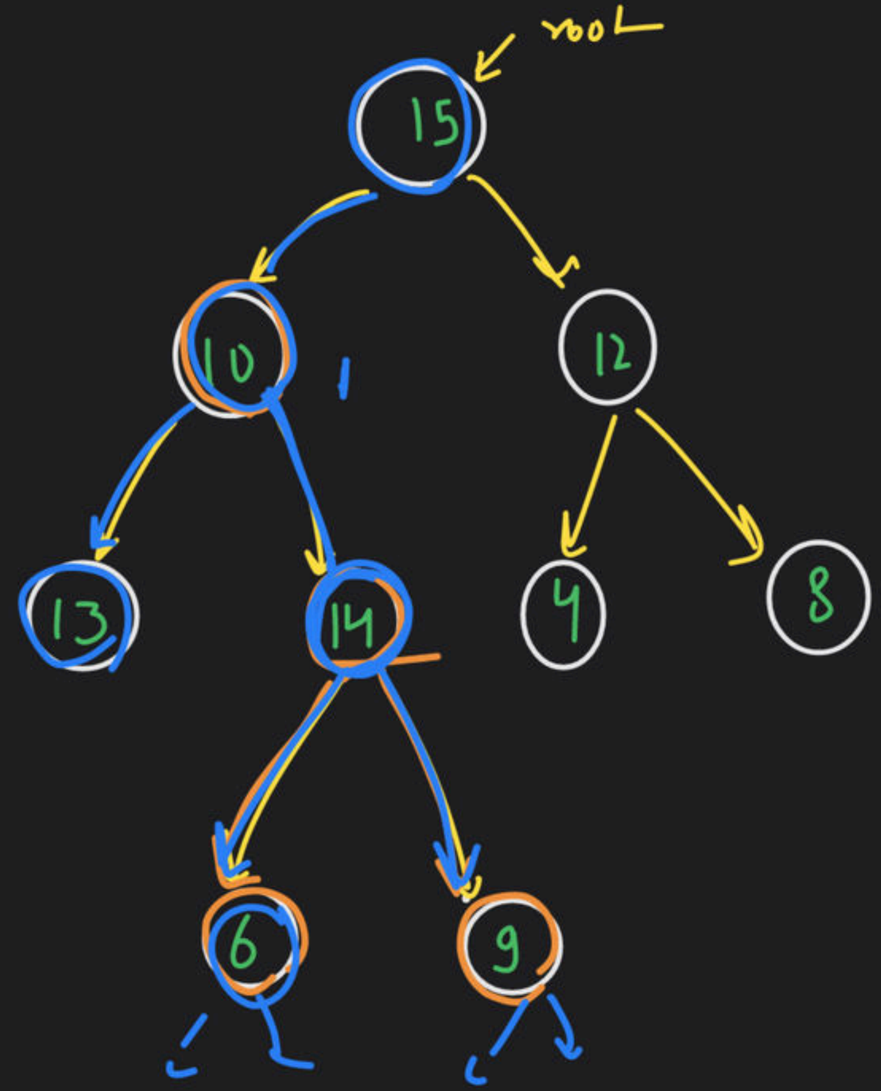
Solving Hard Questions - Level 3 & Doubt Clearing Session - LIVE

Special class

Vishal Pachauri • Sept 25, 2022

→ Sum of nodes within K distance from target



target = 14
K = 2

target = 14
K = 1

= 14 + 6 + 9 + 10 + 13 + 15

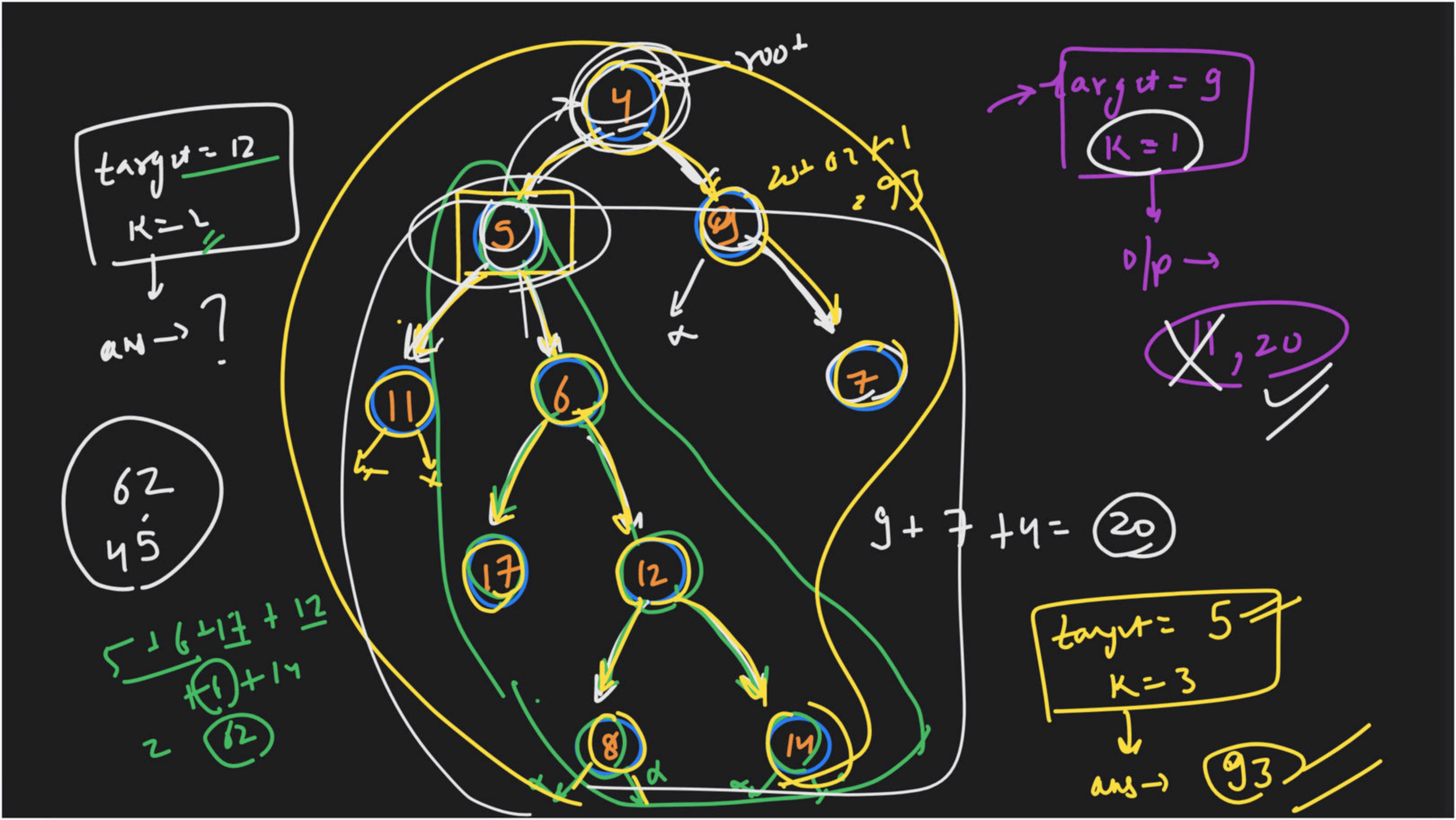
= 20 + 19 + 21

= 39 + 25

= 64

→ 6 + 9 + 10 + 14

15 + 24 = 39


target = 12
K = 2
ans → ?
root
4
5
9
11
6
7
17
12
8
14
target = 9
K = 1
o/p →
11, 20
9 + 7 + 4 = 20
target = 5
K = 3
ans →
93

Algo:-

1. Create & node - parent mapping -> O(n)
2. Find target -> O(n)
3. traverse K steps

Rec
- left part
- right part
- parent

-> O(n)

T.C -> O(n)

① Create node – parent map:

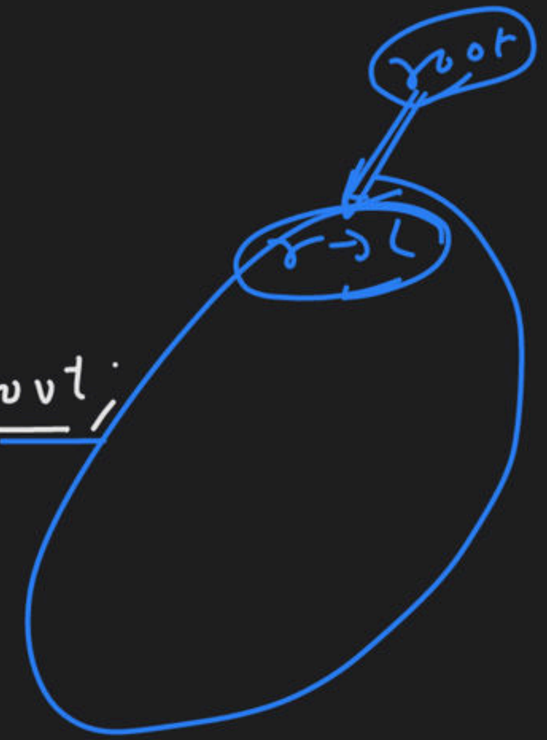
unordered_map < int, node* > mp;
node*, node*

Dummy Node

parent (root) -> (-1)

```
void traverse ( root , mp parent )
{
    if (root == NULL)
        return;

    if (root -> left != NULL)
    {
        parent (root->left) = root;
        traverse (root -> left);
    }
```



```
if (root->right != NULL)
{
    parent[root->right] = root;
    traverse(root->right, parent);
}
```

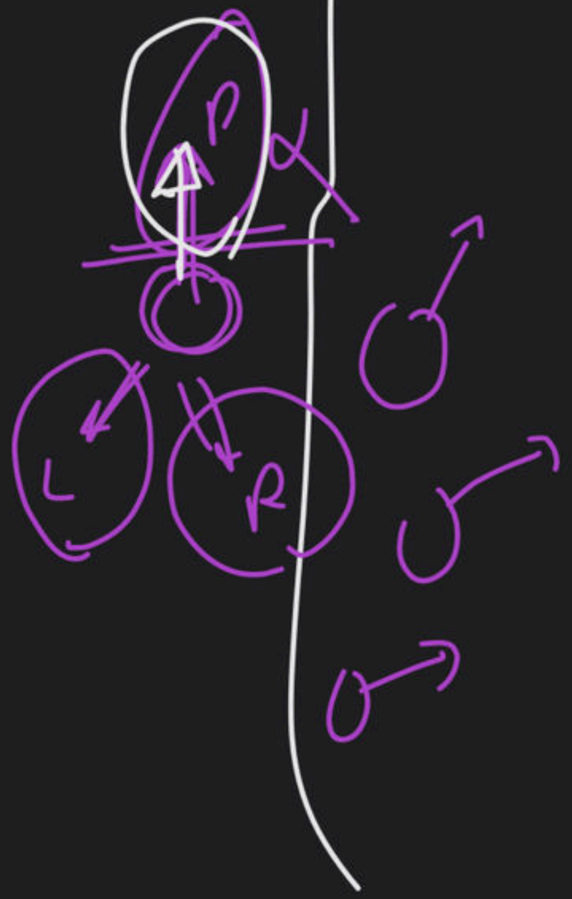
```
Node* search(root, target)
{
    if (root == NULL)
        return NULL;

    if (root->data == target)
        return root;

    node* leftAns = search(root->left, target)
    if (leftAns != NULL)
        return leftAns

    return search(root->right, target)
}
```

④ traverse in each direction [3 dir^s] for K steps




```
void
{
}
```

```
findAns ( target, K, ans, map<Node*, bool> vis)
```

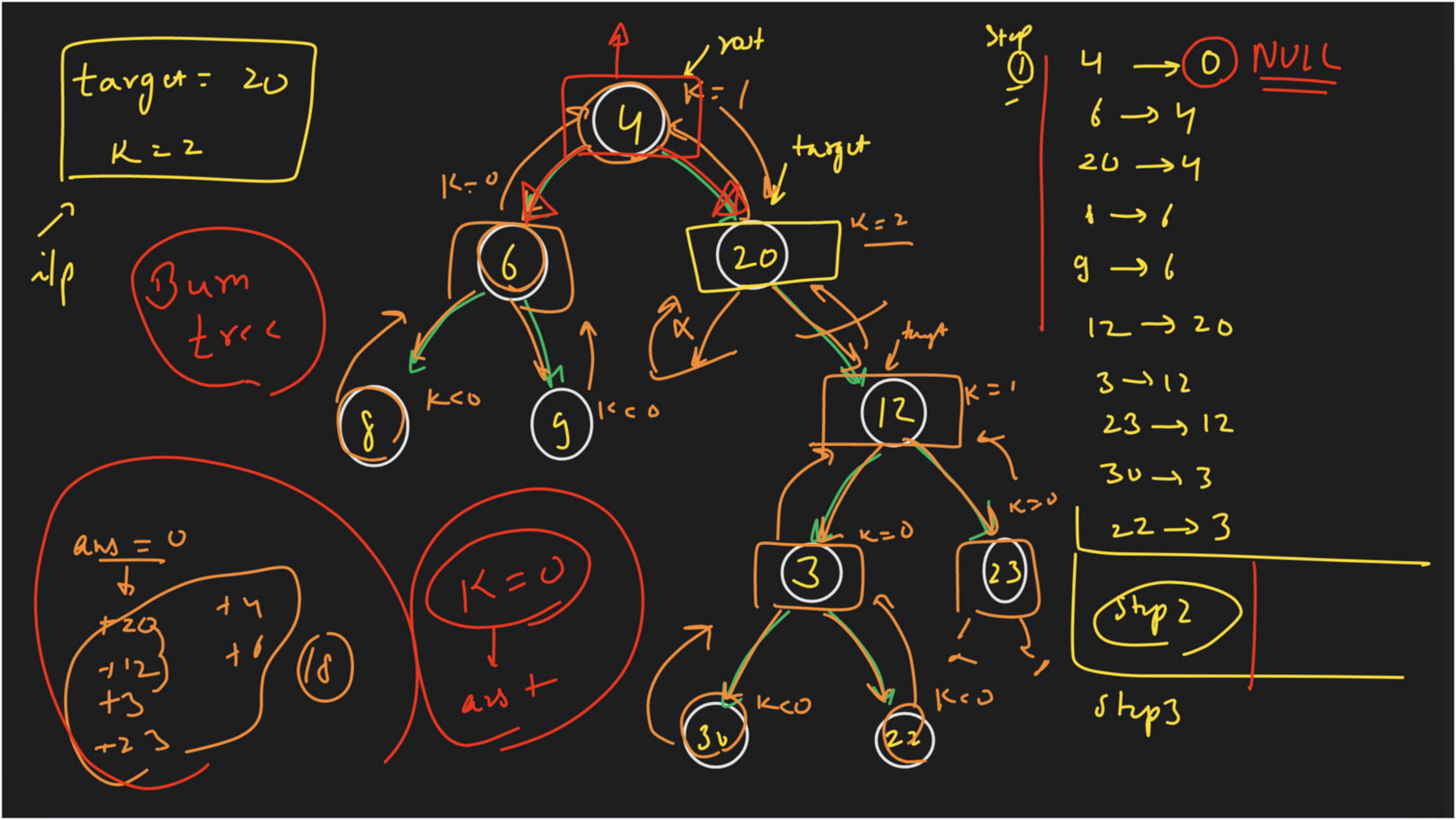
```
if ( K < 0)
    return ;
```

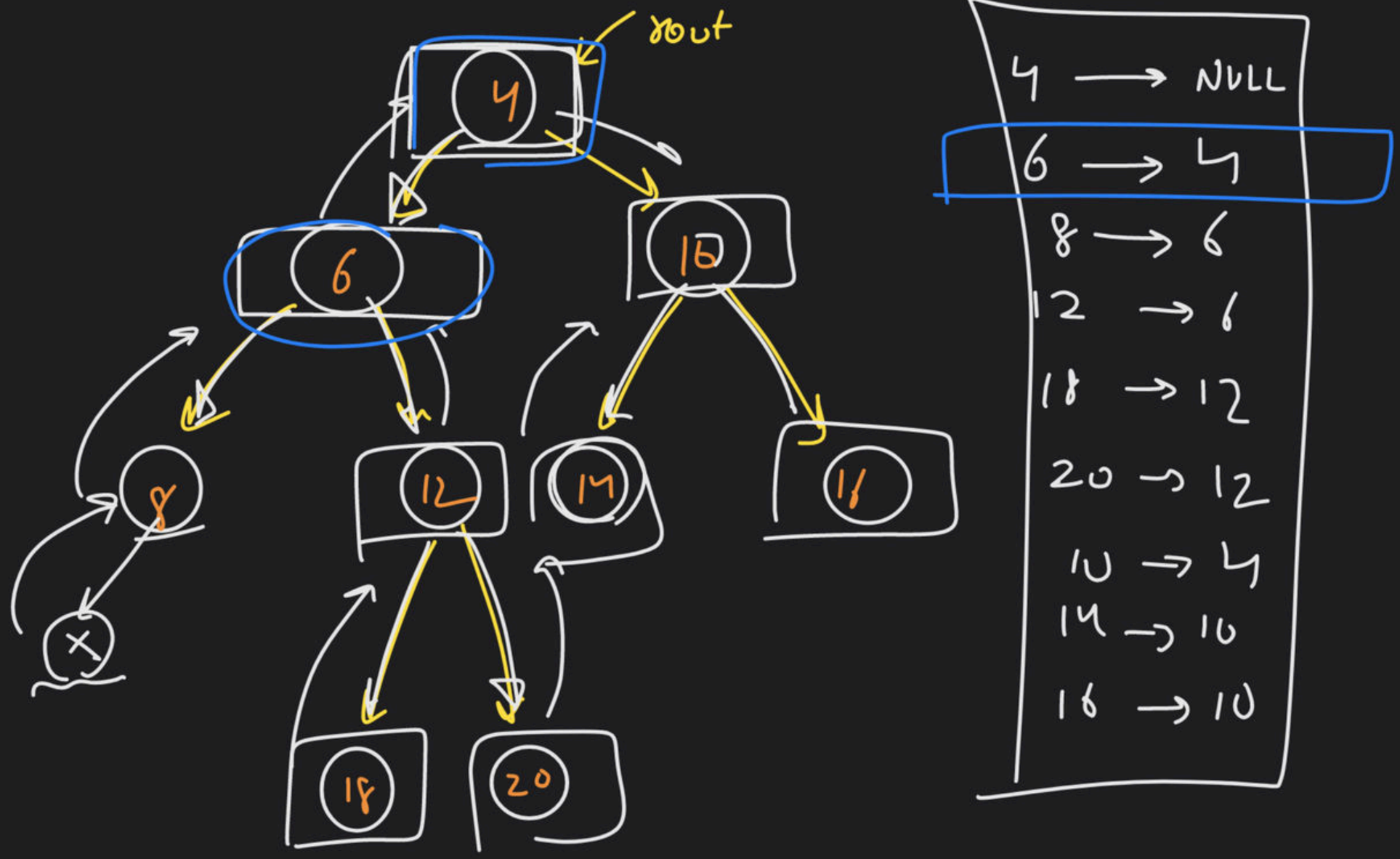
```
if (vis[target])
    return;
vis[target] = true;
```

```
ans = ans + root -> data;
```

```
if (target == NULL)
    return ;
```

```
findAns ( target -> left , K-1 , ans , vis)
findAns ( target -> right , K-1 , ans , vis)
findAns ( parent(target) , K-1 , ans , vis )
```


target = 20
K = 2
i/p
Bum tree
root
K = 1
target
4
K = 0
6
20
K = 2
8
K < 0
9
K < 0
temp
12
K = 1
K = 0
3
23
K > 0
3b
K < 0
22
K < 0
ans = 0
+20
+4
-12
+6
+3
18
+23
K = 0
ans +
Step
1
4 → 0 NULL
6 → 4
20 → 4
1 → 1
9 → 6
12 → 20
3 → 12
23 → 12
30 → 3
22 → 3
step2
step3


root
4
6
10
8
12
14
11
X
18
20
4 → NULL
6 → 4
8 → 6
12 → 6
18 → 12
20 → 12
10 → 4
14 → 10
16 → 10

(1) Minimise operations to reduce A & B to 1

(a) by decrementing by 1

(b) or divide a by b or b by a

HARD

Google

Amazon

A = 13

B = 5

A = 1

B = 1

a/b

a%b = 0

b%a = 0

a == b → 1 → 1 1

option → a-1 or b-1

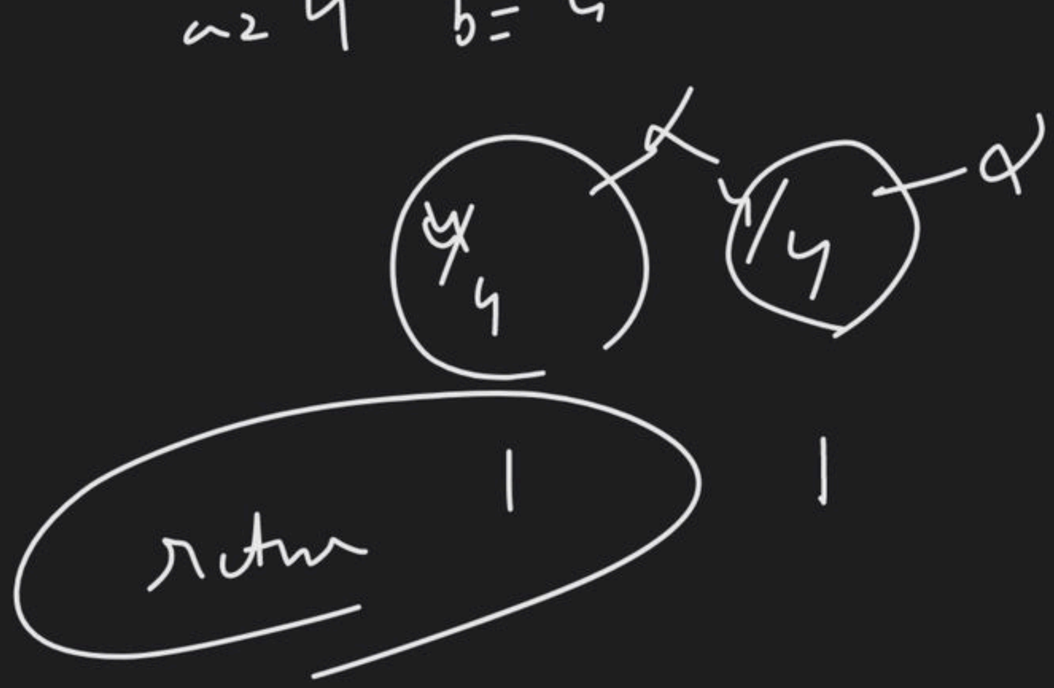
option → a/b

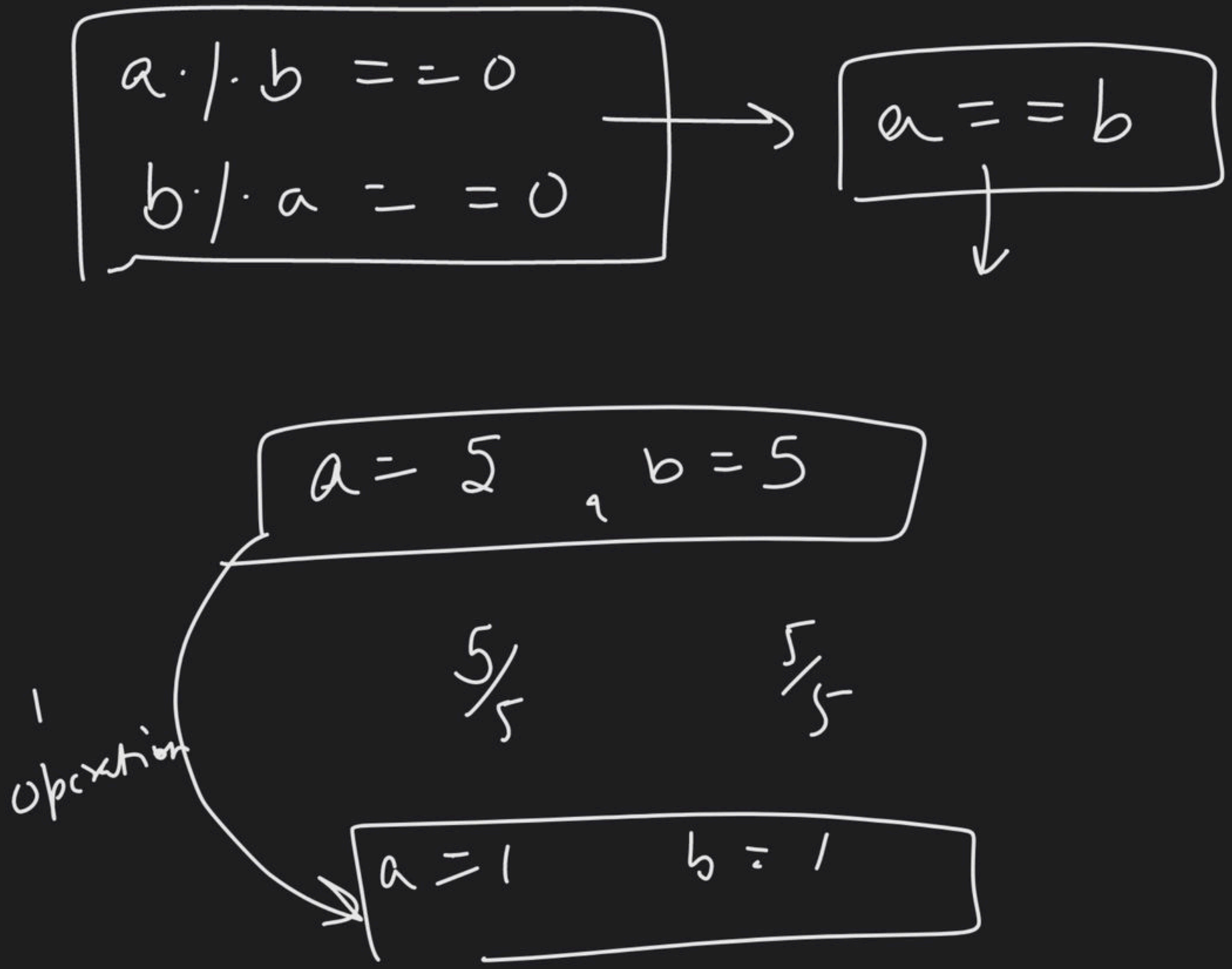
opti → b/a

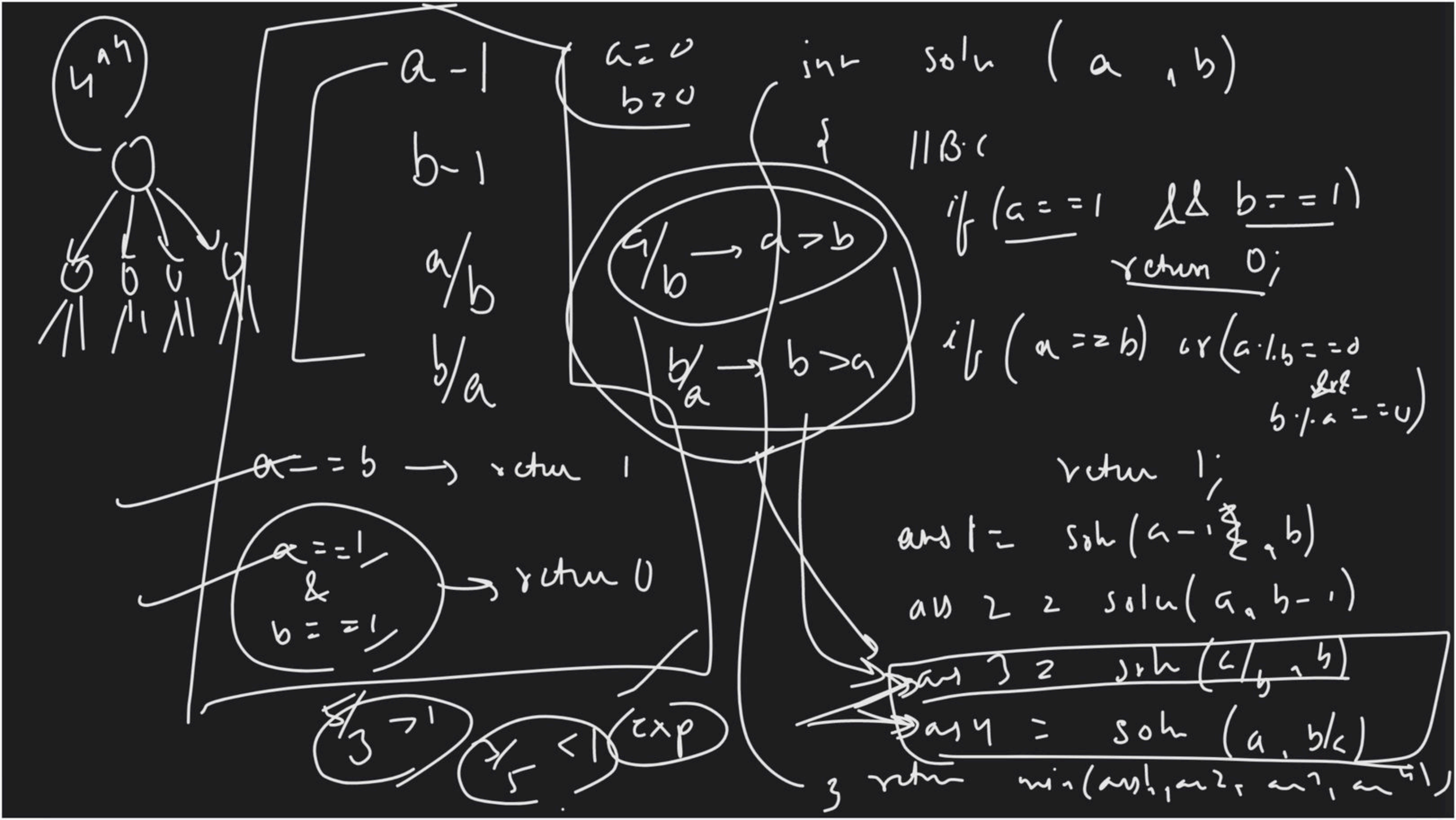
optiu → a/b & b/c

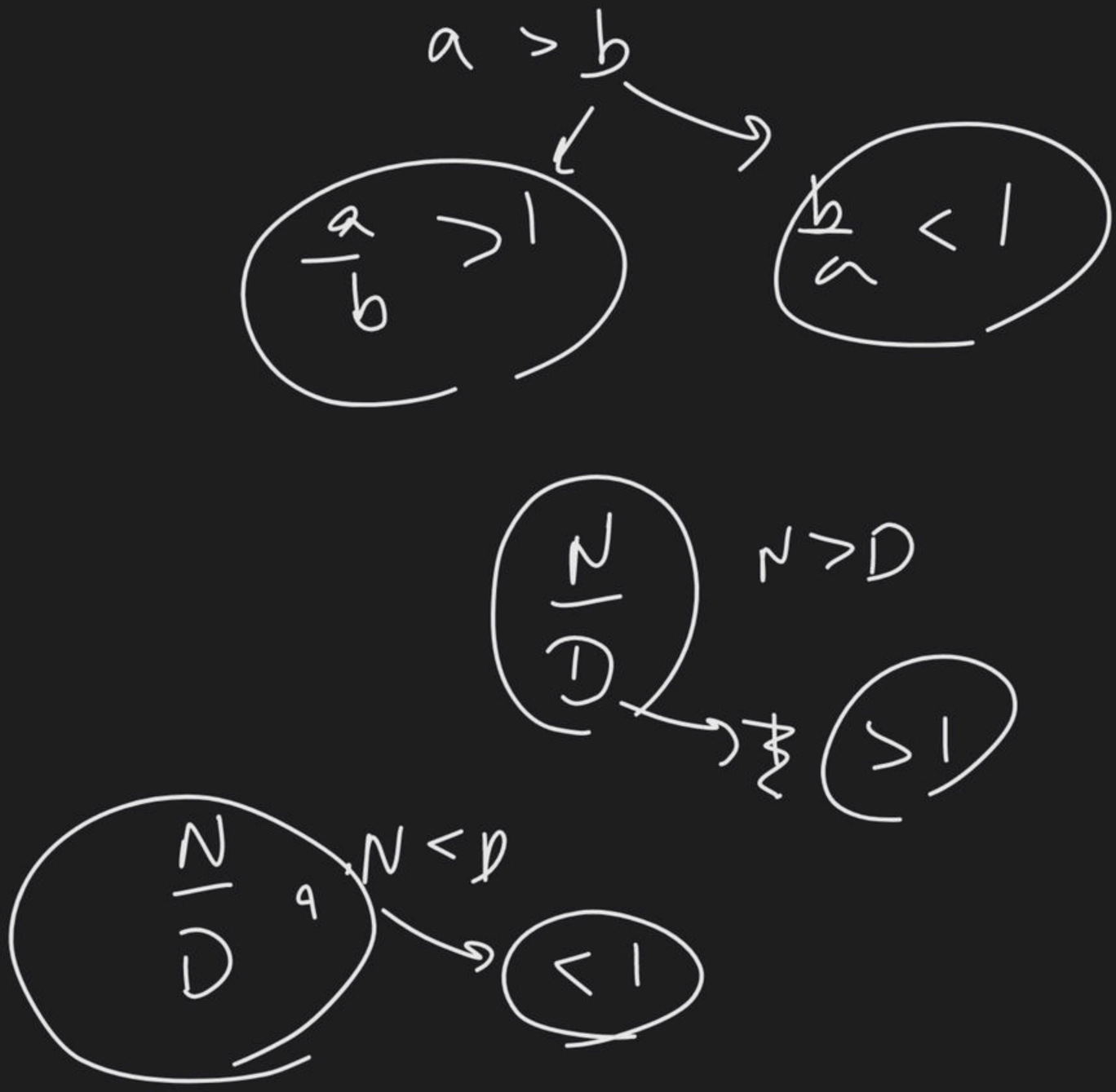
12 → 12 → 11

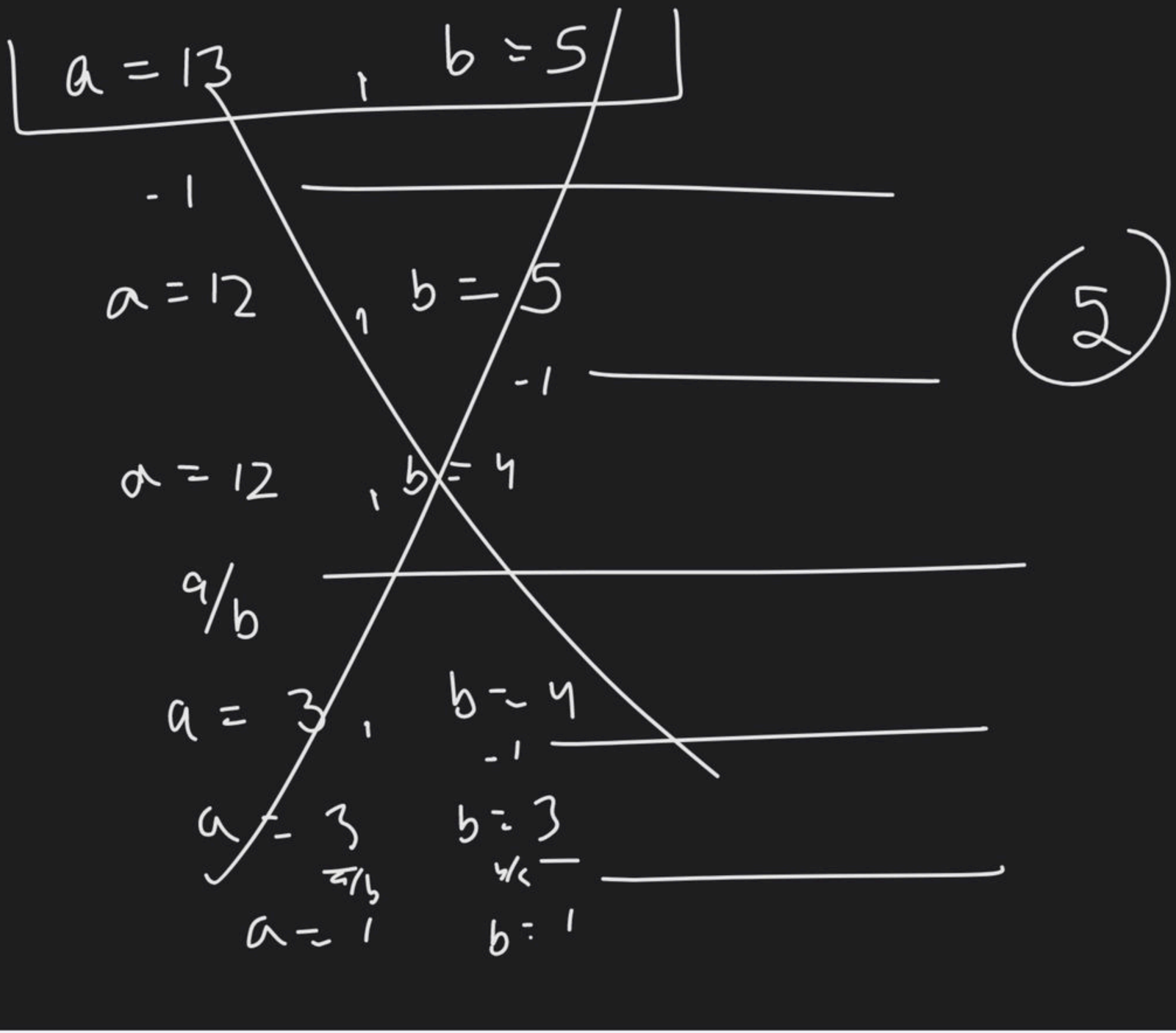
5 4 4


a=4 b=4
return


a % b == 0
b % a == 0
a == b
a = 5 , b = 5
5/5
5/5
1 operation
a = 1 b = 1


a-1
b-1
a/b
b/a
a==0
b>0
a/b → a>b
b/a → b>a
a==b → return 1
a==1 & b==1 → return 0
a/b >1
a/b <1
exp
int soln (a, b)
{ //B.C
if (a==1 && b==1)
return 0;
if (a==b) or (a%b==0 but b%a==0)
return 1;
ans1 = soln(a-1, b)
ans2 = soln(a, b-1)
ans3 = soln(a/b, b)
ans4 = soln(a, b/a)
return min(ans1, ans2, ans3, ans4)


a > b
a/b > 1
b/a < 1
N/D
N > D
N/D
N < D
< 1



(2)

① decrement either a or b by 1 — a-1, b-1

② a/b → rem → 0

a/b b/a a%b & b%a

if (a==1 && b==1) return 0;

a, c →

if (a%b==0 && b%a==0) return 1

ans1 = (a-1, b)

ans2 -1 (a, b-1)

if (a%b == 0)

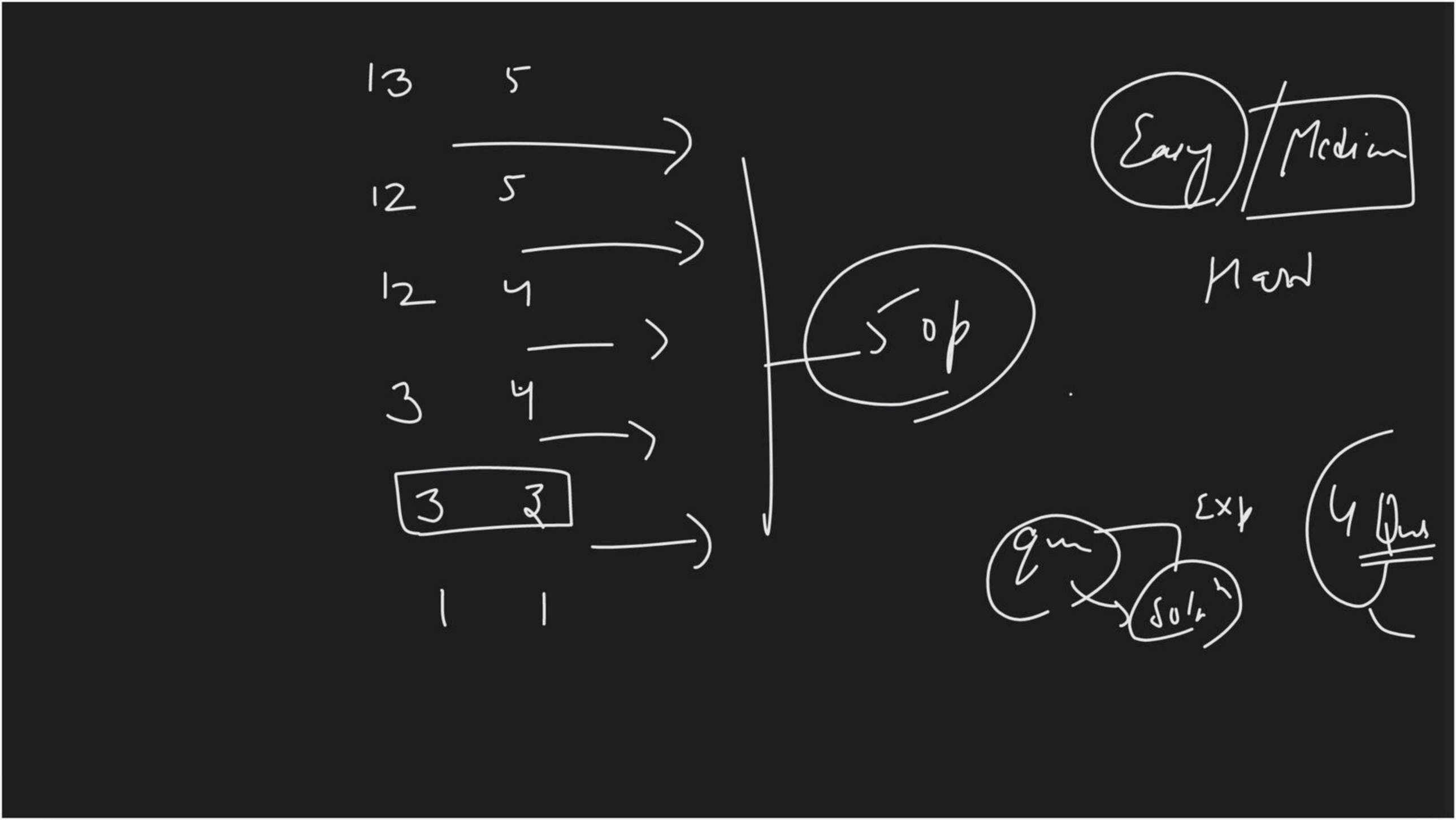
ans3 = (a/b, b)

if (b%a == 0)

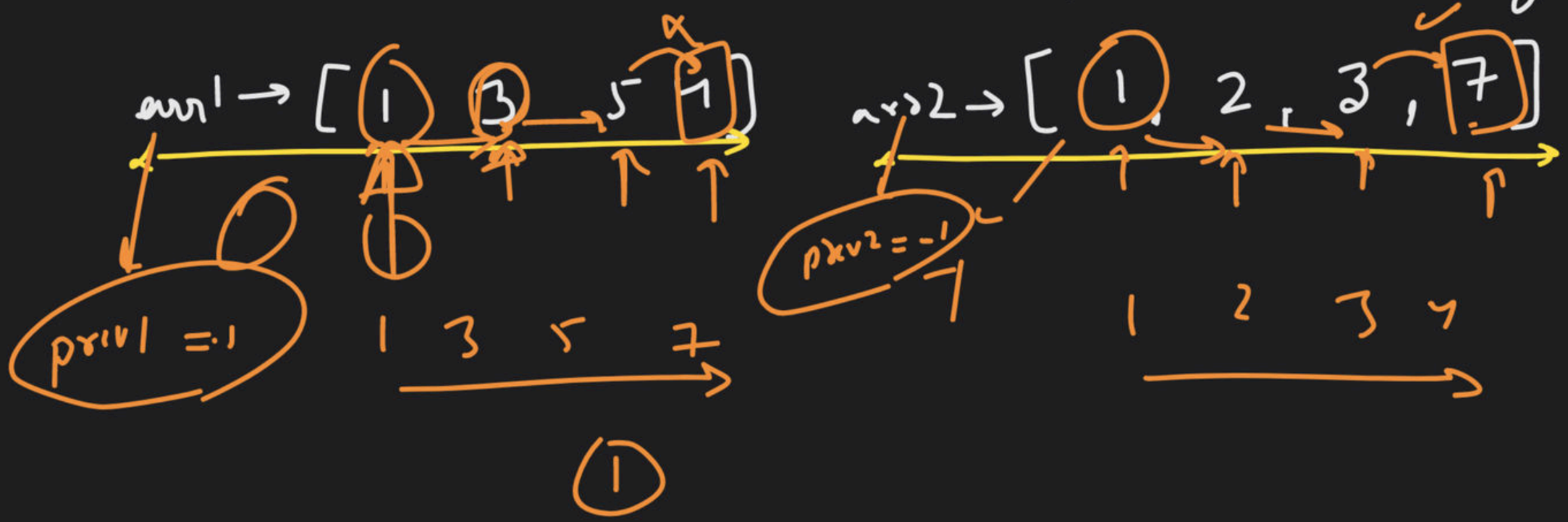
ans4 = (a, b/a)

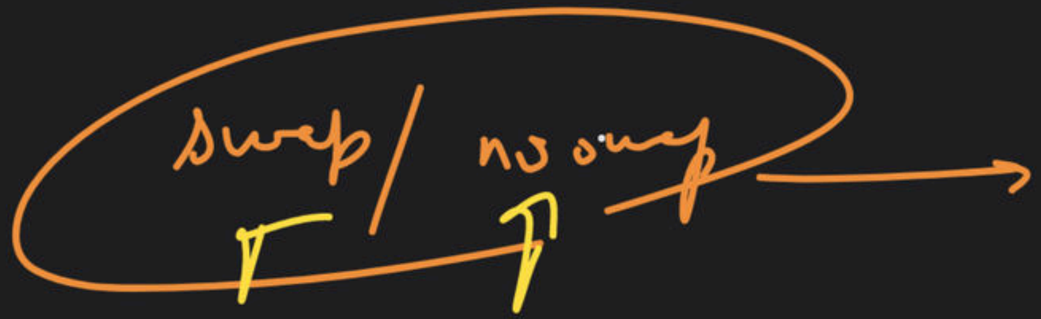
findAns = min (ans1, ans2, ans3, ans4)

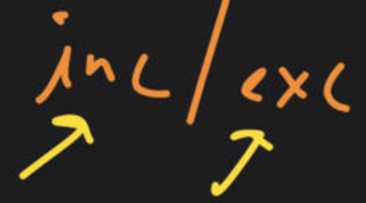
3 return findAns

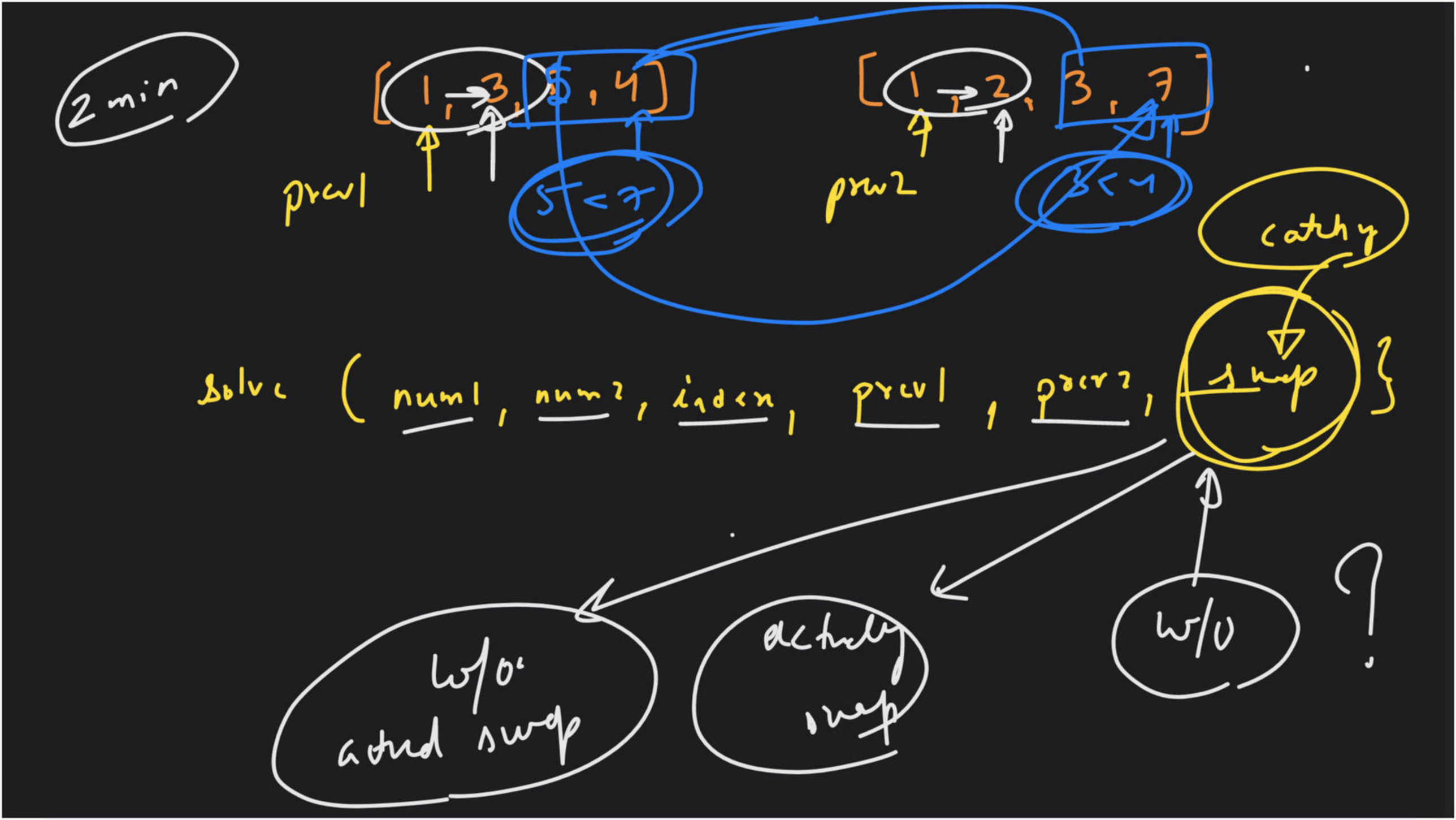

13 5
12 5
12 4
3 4
3 3
1 1
Sop
Easy
Medium
Hard
Exp

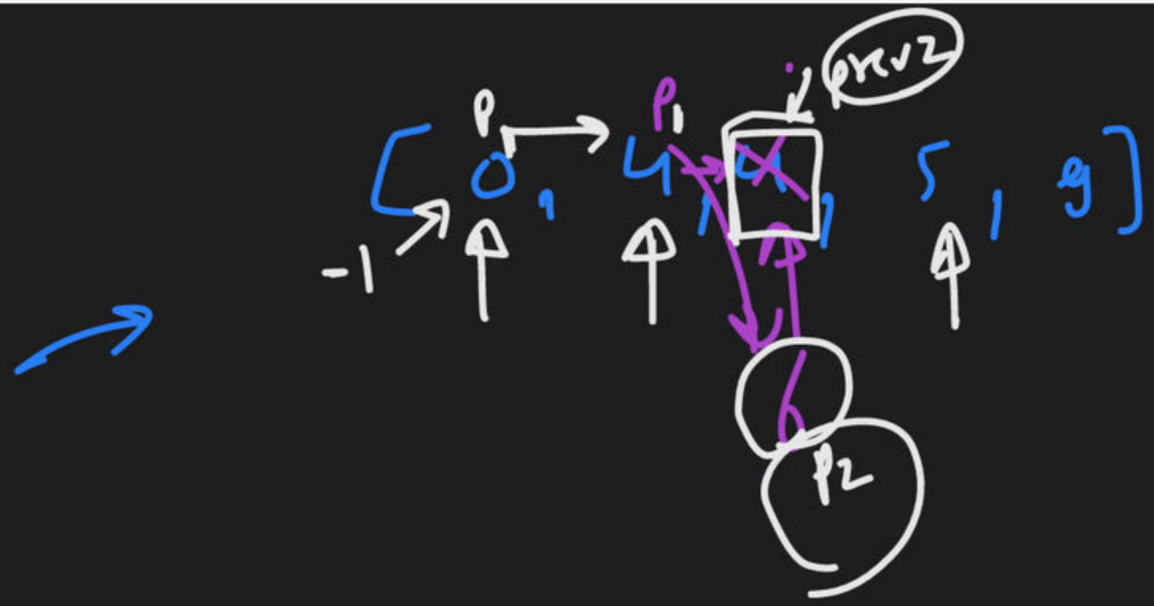
③ Minimum Swaps to make Sequence increasing

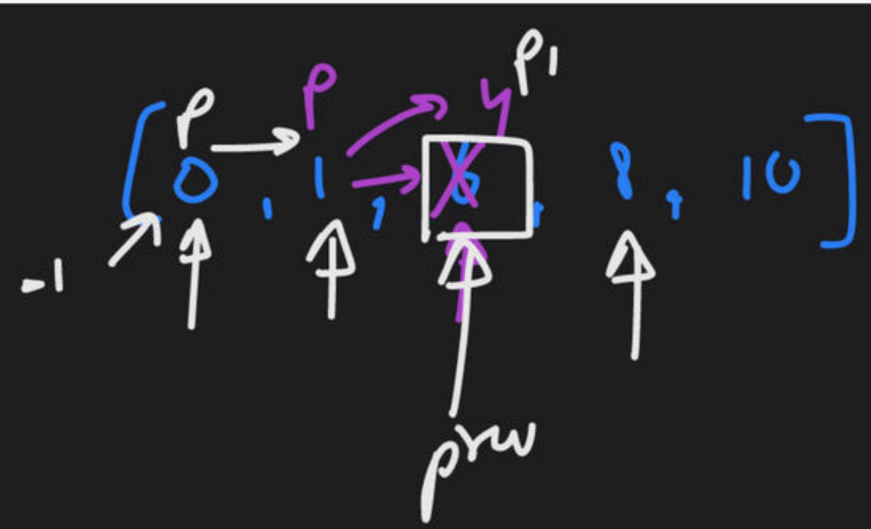


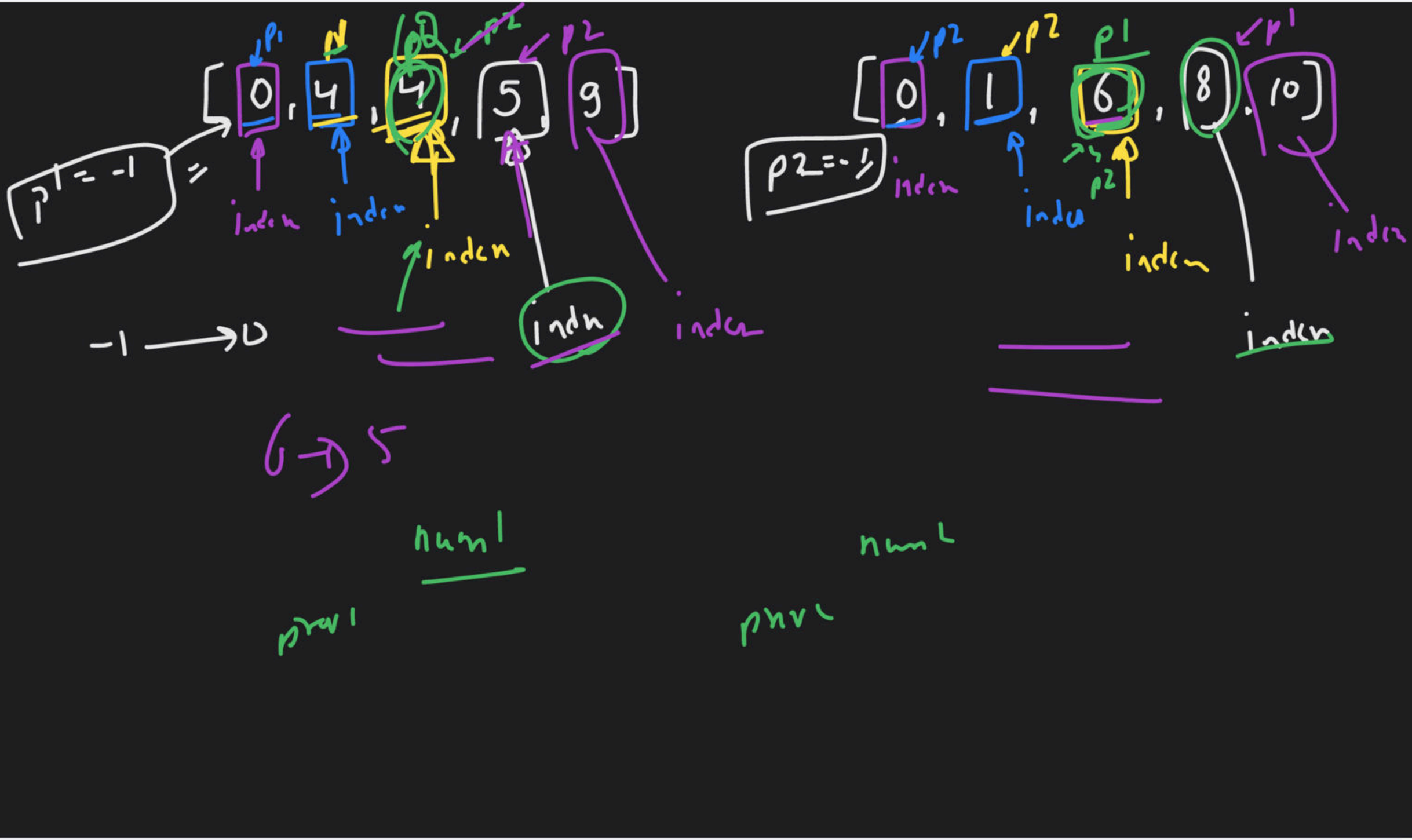


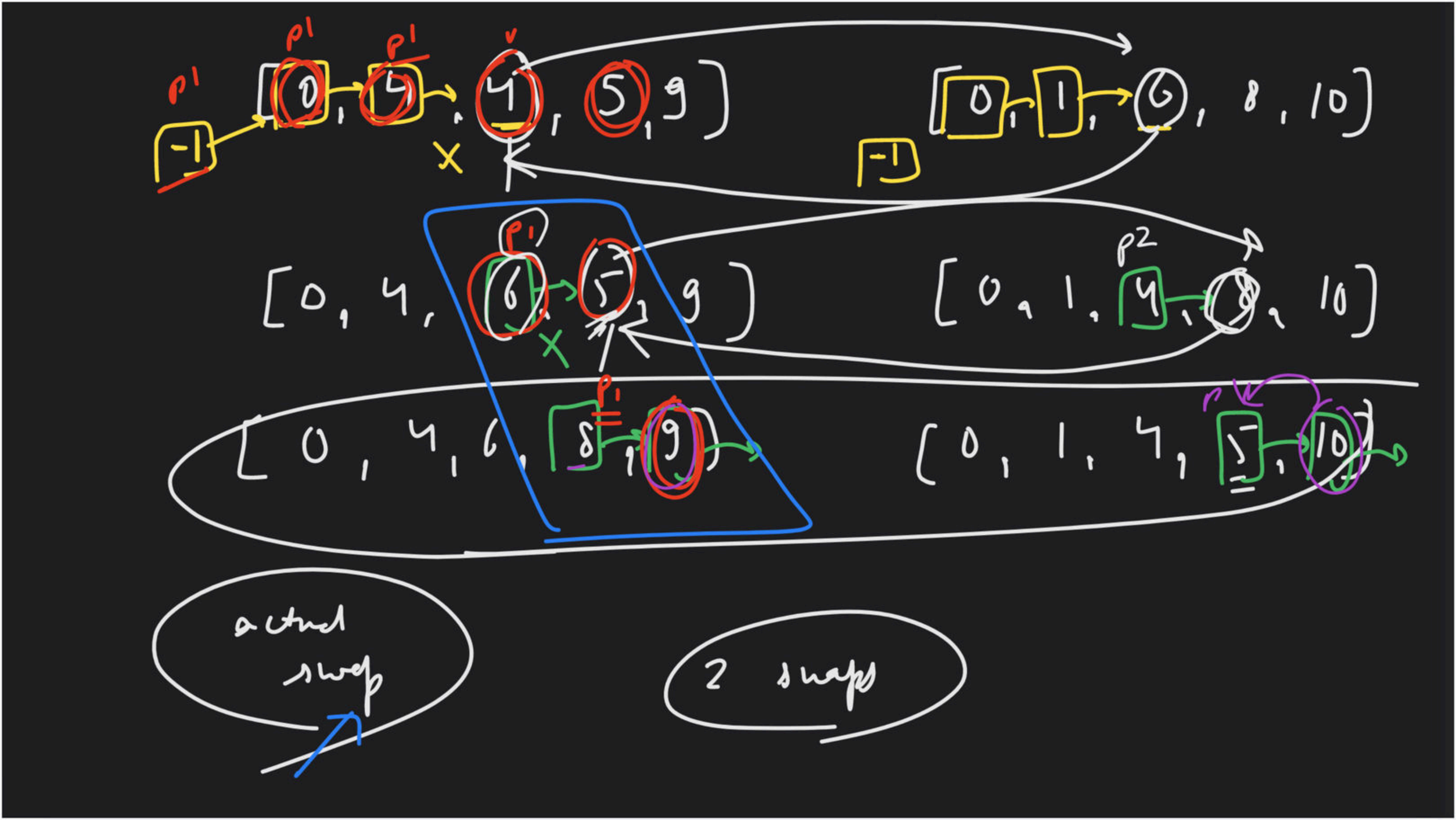


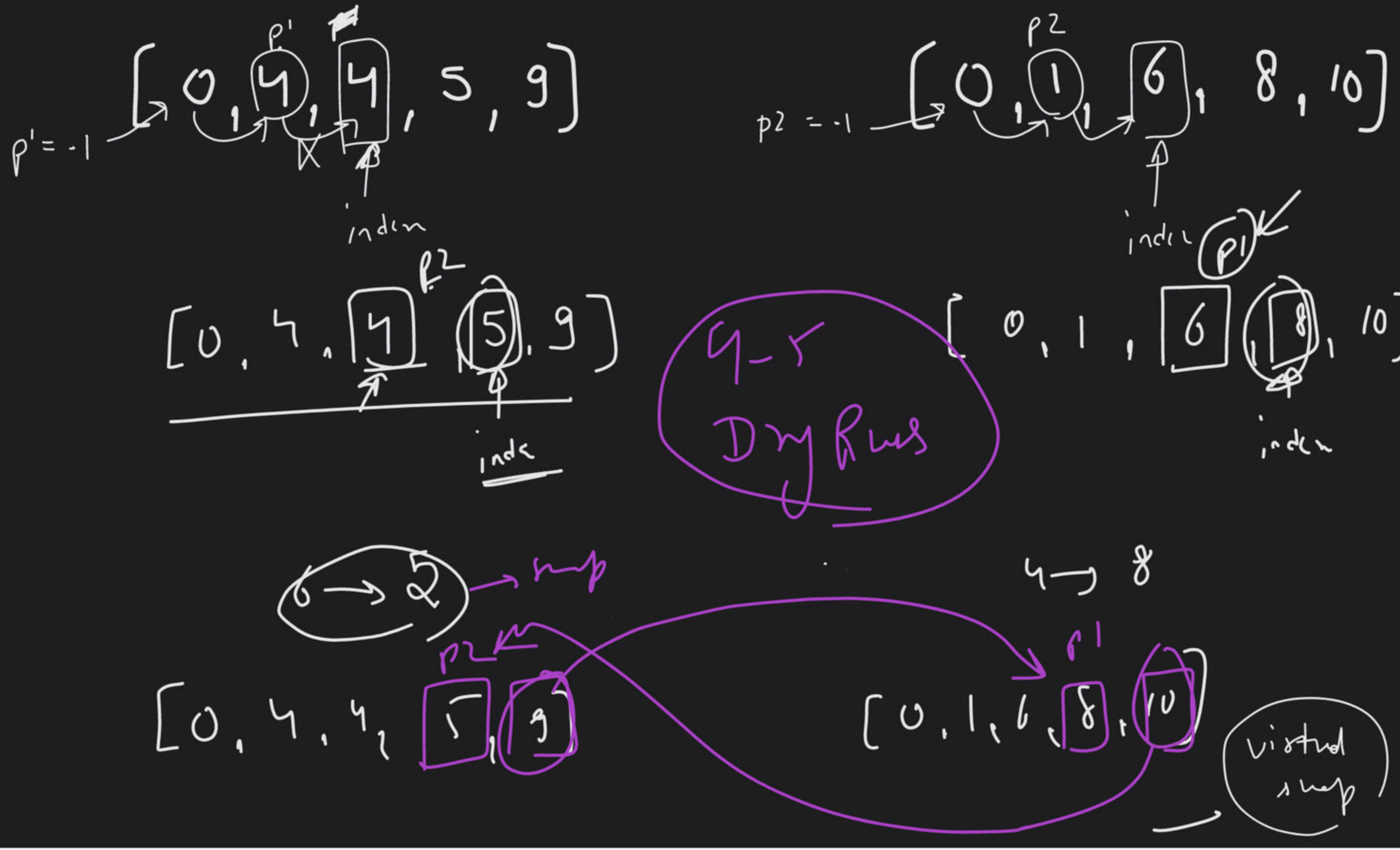

2 min
[1, 3, 5, 4]
[1, 2, 3, 7]
prev1
prev2
5 < 7
3 < 4
catchy
solve (num1, num2, index, prev1, prev2, swap)
w/o actual swap
actually swap
w/o
?


p
p1
prv2
-1
0
1
4
5
9
6
p2


p
p
p1
-1
0
1
8
10
prv


p1
p2
p2
[0, 4, 4, 5, 9]
p1 = -1
index
index
index
index
index
-1 → 0
6 → 5
num1
prev1
[0, 1, 6, 8, 10]
p2
p2
p1
p1
p2 = -1
index
index
p2
index
index
index
numL
prevL


p1
p1
p1
v
[0, 4, 5, 9]
-1
X
[0, 1, 6, 8, 10]
-1
p1
[0, 4, 6, 5, 9]
X
p2
[0, 1, 4, 8, 10]
p1
[0, 4, 6, [8, 9]]
[0, 1, 4, [5, 10]]
actual swap
2 swaps


p'
p'=-1
0, 4, 4, 5, 9
index
p2
p2=-1
0, 1, 6, 8, 10
index
p1
0, 4, 4, 5, 9
index
0, 1, 6, 8, 10
index
q-r
DryRun
0 → 2
swap
4 → 8
p2
p1
0, 4, 4, 5, 9
0, 1, 6, 8, 10
visited
swap

BT

14-15

48 Qns

Dry Run

17 Jan

H/w → Reducing Dishes → Electrode

T.C / S.C

Rec → Revision