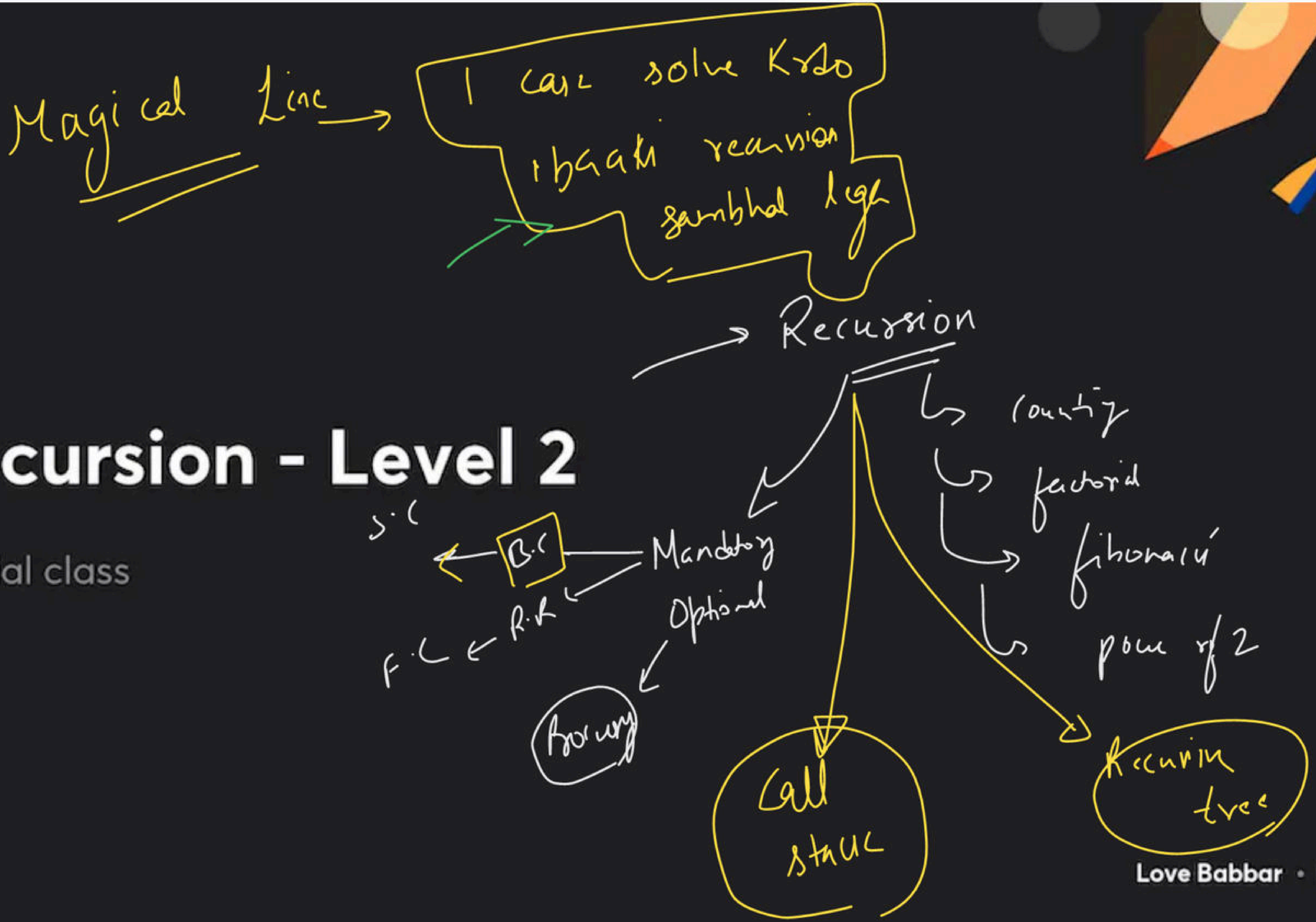
Recursion - Level 2

Special class

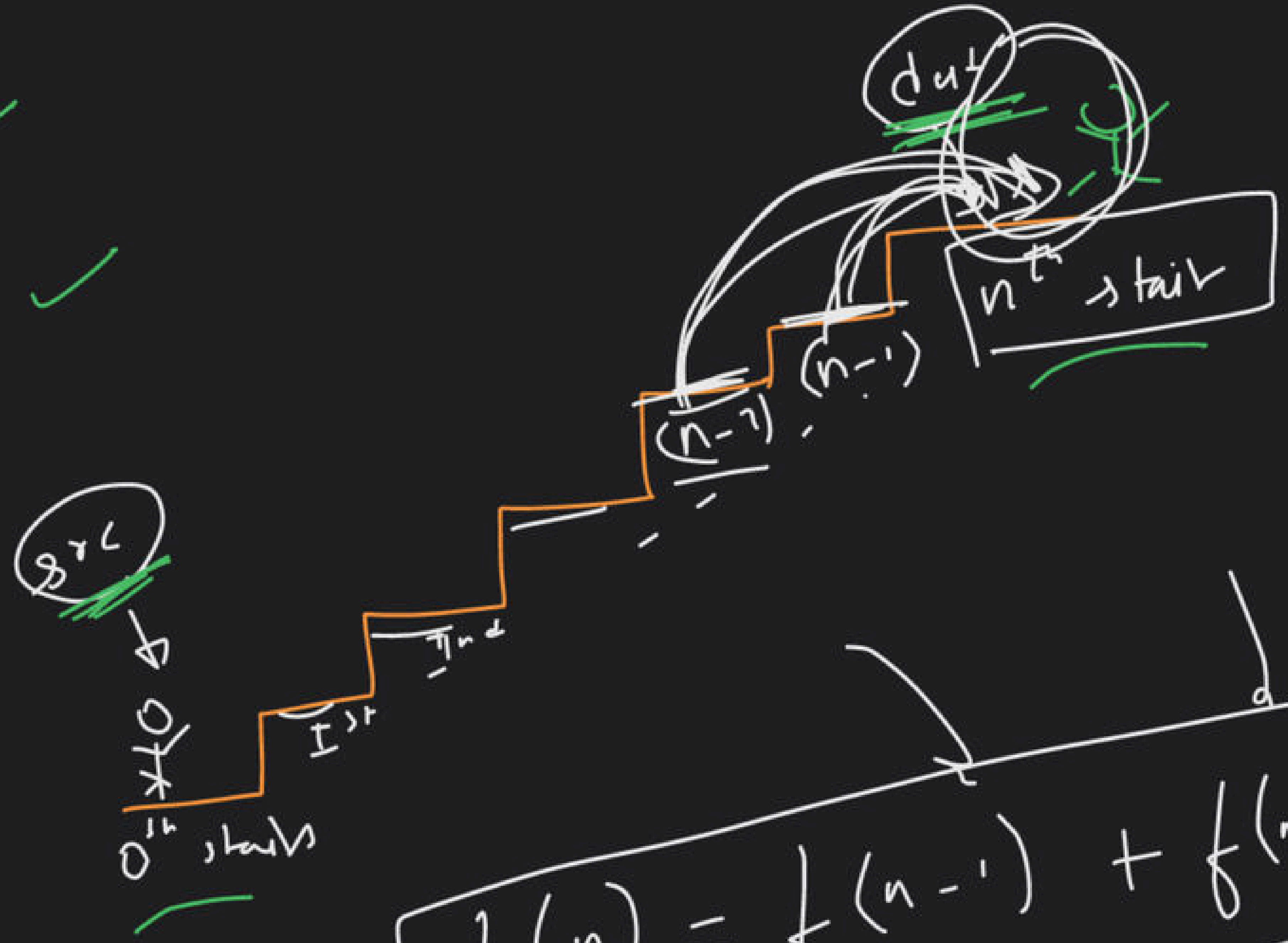


Love Babbar • Mar 11, 2023

2min

steps allowed
- 1 stair at a time ✓
- 2 stair at a time ✓

find no of ways to reach nth stair

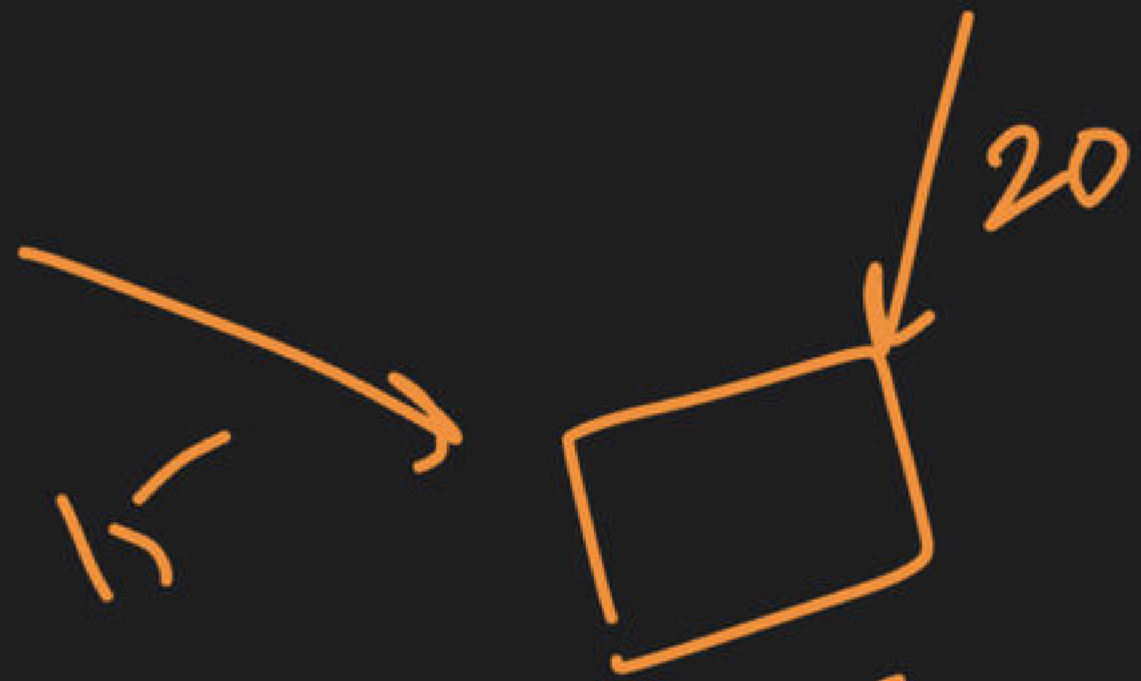


$$\boxed{f(n) = f(n-1) + f(n-2)}$$

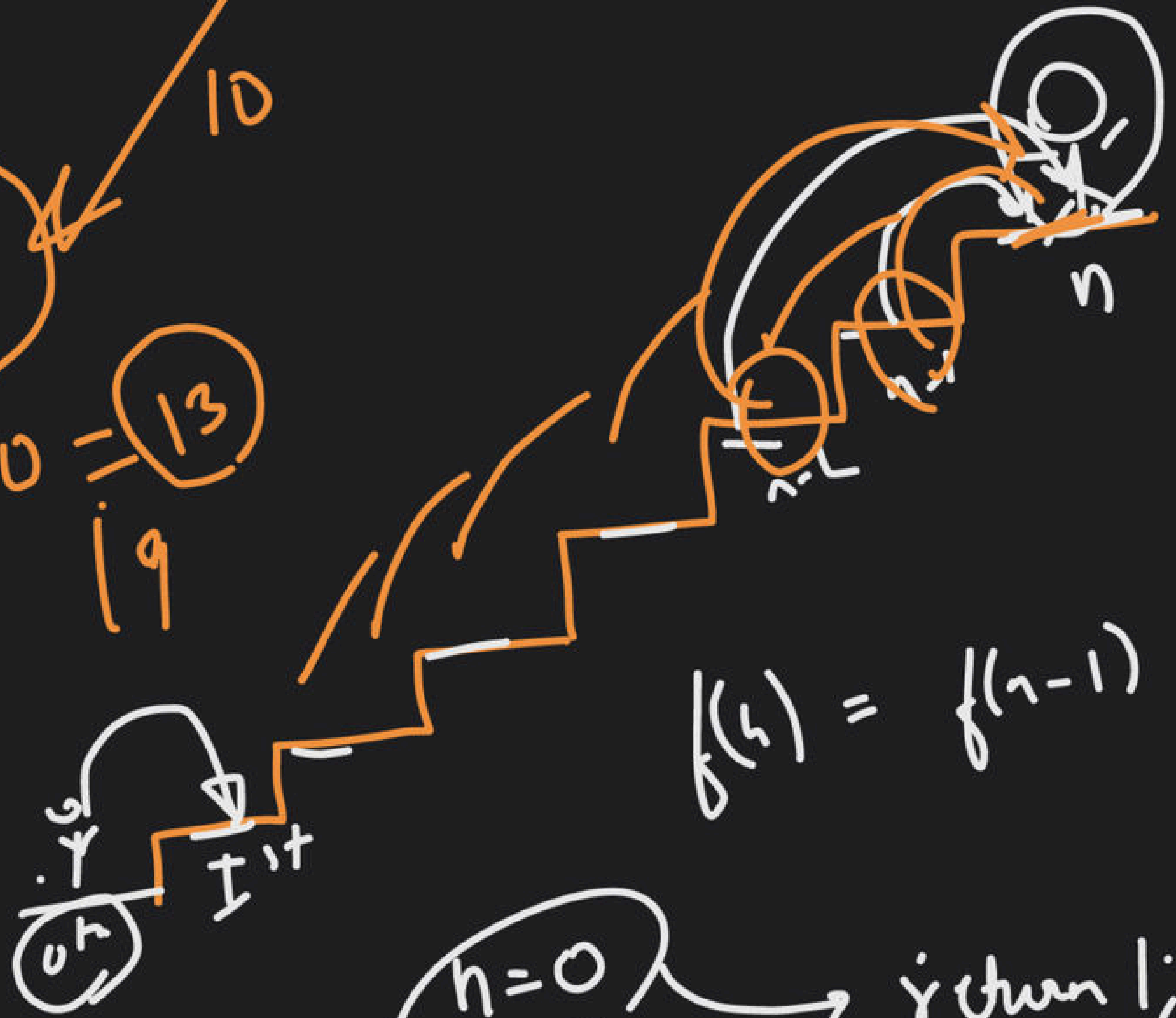
~~Total no of ways to reach nth stair~~ = -

$3+10 = \boxed{13}$

19



$15+20 \rightarrow 35$



$f(n) = f(n-1) + f(n-2)$

$n=0 \rightarrow$ return 1;

$n=1 \rightarrow$ return 1;

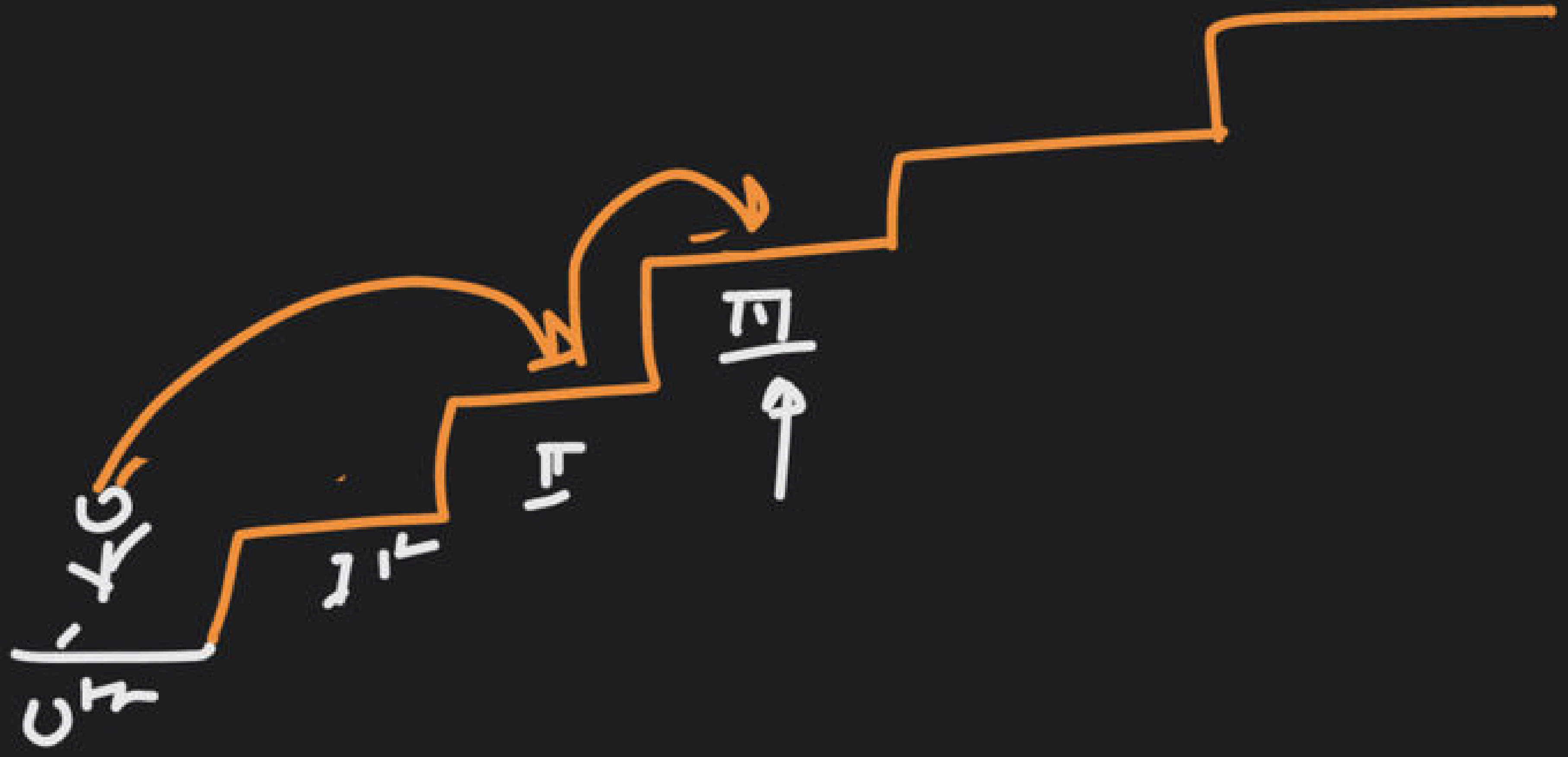
$$\underline{\underline{f(n)}} = f(n-1) + f(n-2)$$

$$\begin{tikzcd}
 & & 10 \arrow[dl] \arrow[dr] & \\
 & 9 \arrow[dl] \arrow[d] & & 8 \\
8 & 7 & &
\end{tikzcd}$$

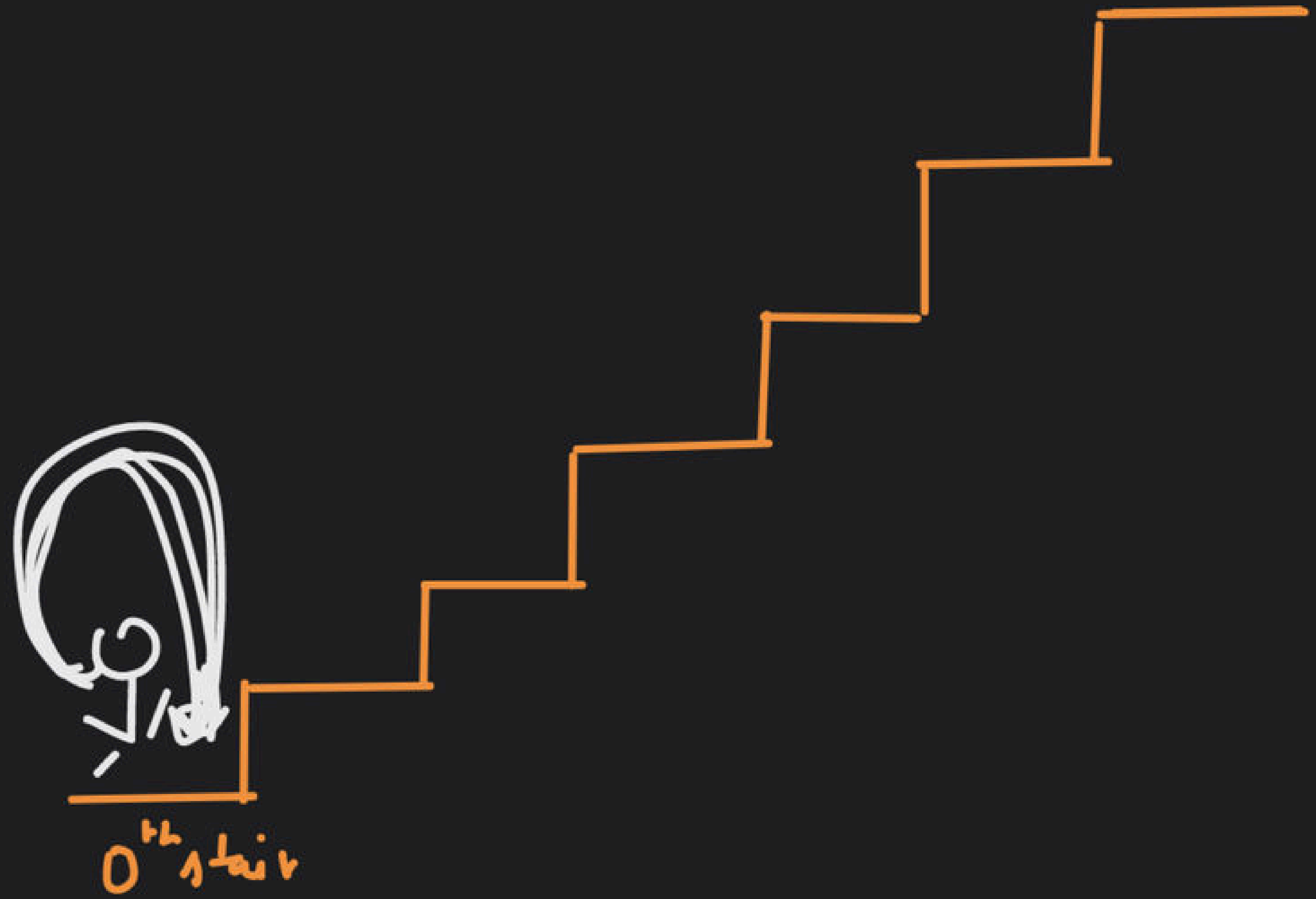
1, 1, 1

1, 2

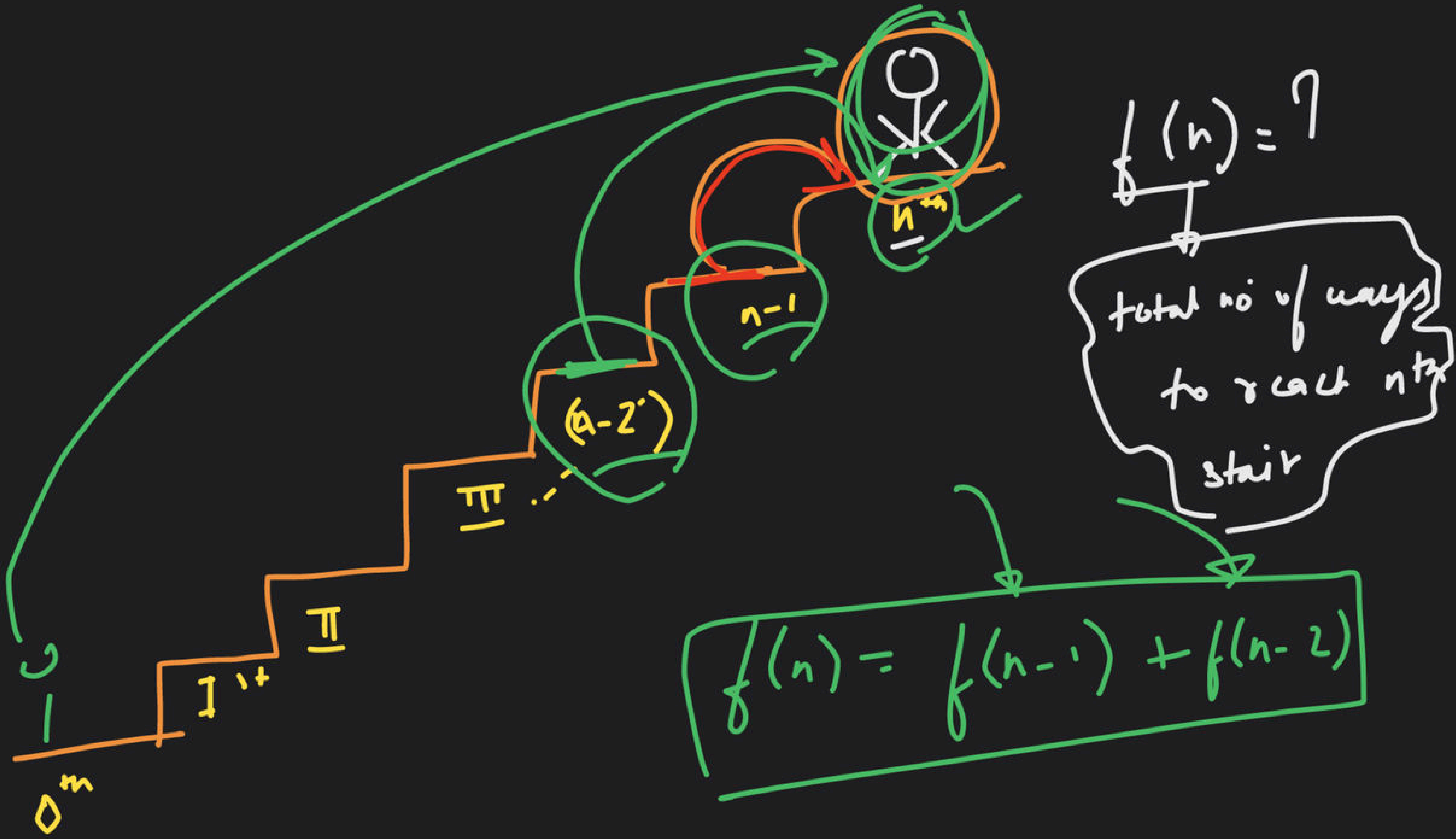
2, 1



0th stair
↳ ways = 0



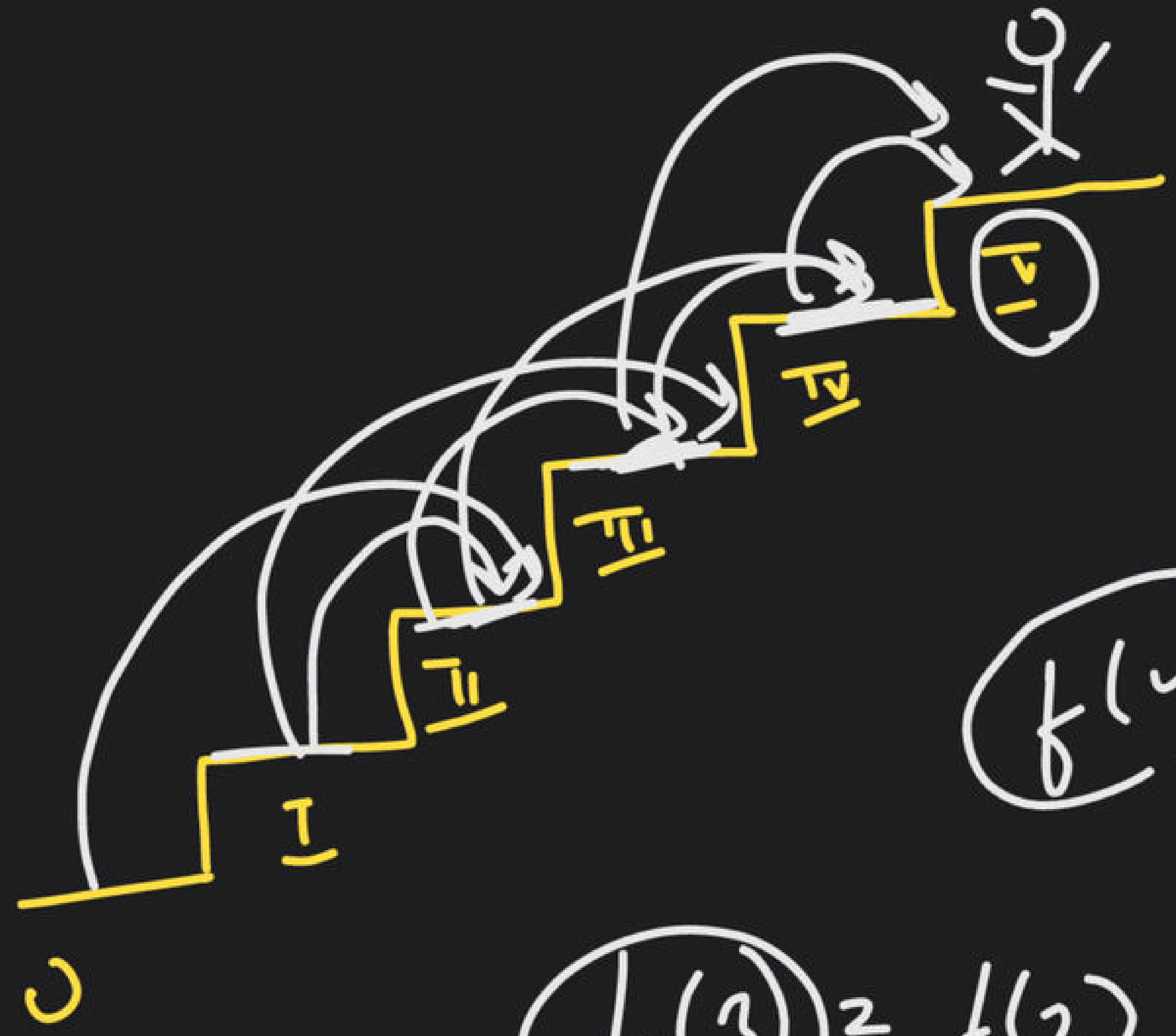
0th
↳ way = 1



$$\boxed{f(n) = f(n-1) + f(n-2)}$$

B.C

$$\boxed{\begin{aligned} f(1) &= 1 \\ f(0) &= 1 \end{aligned}}$$

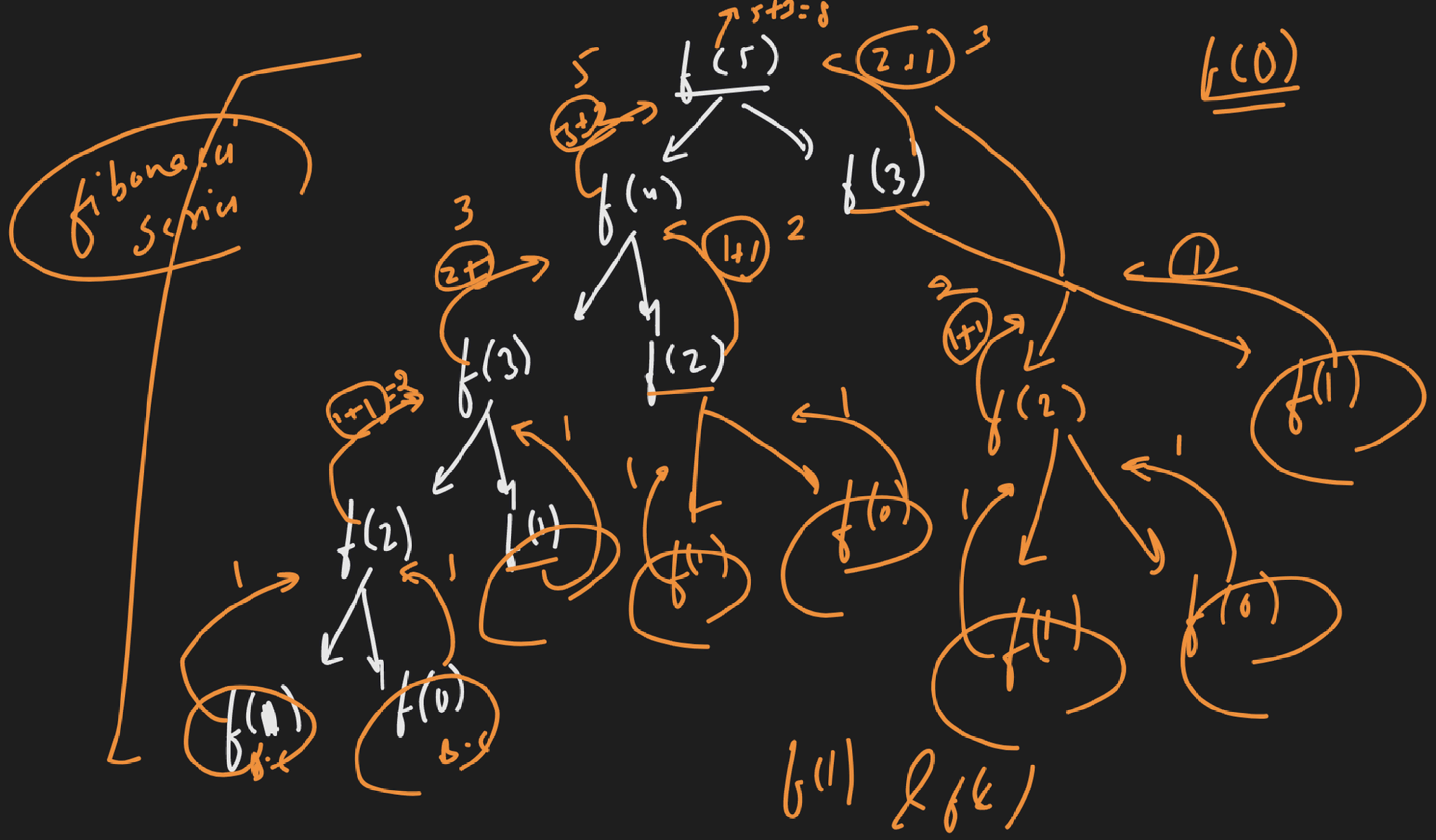


$$f(5) = \underline{f(4)} + \boxed{f(3)}$$
$$= 5 + 3$$
$$= 8$$

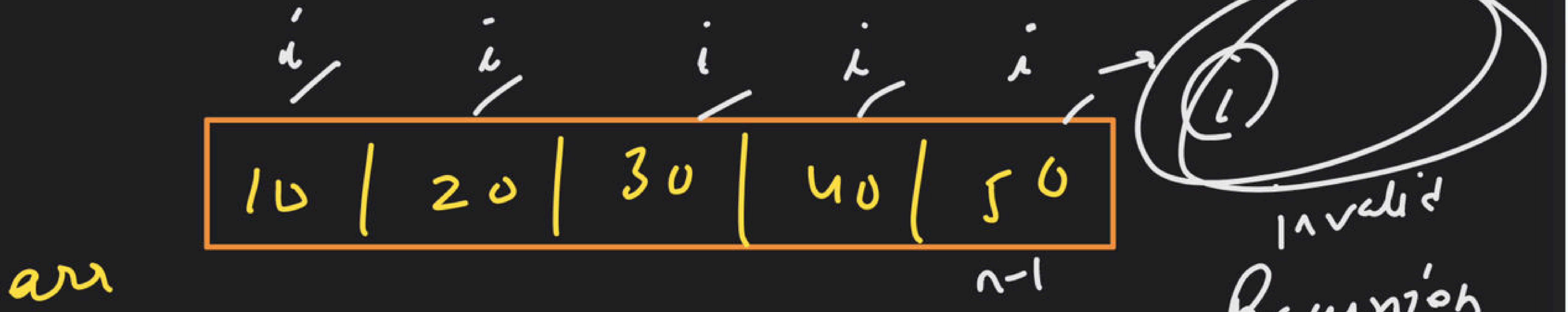
$$\boxed{f(4)} = f(3) + f(2)$$
$$= 3 + 2 = 5$$

$$\boxed{f(3)} = f(2) + f(1)$$
$$= 2 + 1$$
$$= \boxed{3}$$

$$\boxed{f(2)} = f(1) + f(0)$$
$$= 1 + 1 = \boxed{2}$$


fibonacci series
f(5)
5+3=8
f(4)
3+2
5
f(3)
2+1
3
f(2)
1+1=2
f(1)
f(0)
1
1+1
2
f(3)
2+1
3
f(2)
1+1
2
f(1)
f(0)
1
f(0)
f(1) & f(0)

→ Array



Recursion

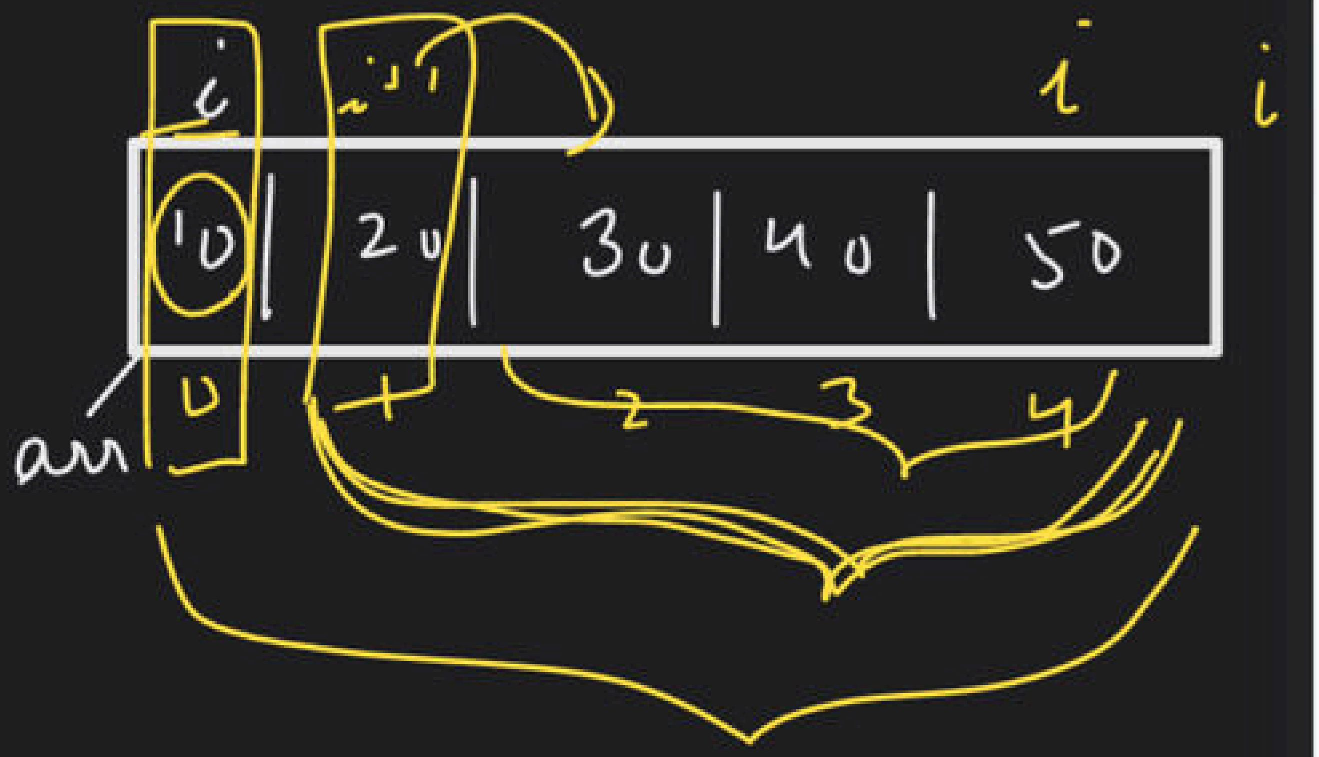
for loop

```
for (int i=0; i<n; i++)
{
      cout << arr[i];

}
```

$$\begin{tikzcd} i>=n \arrow[d] \\ \text{invalid index} \end{tikzcd}$$

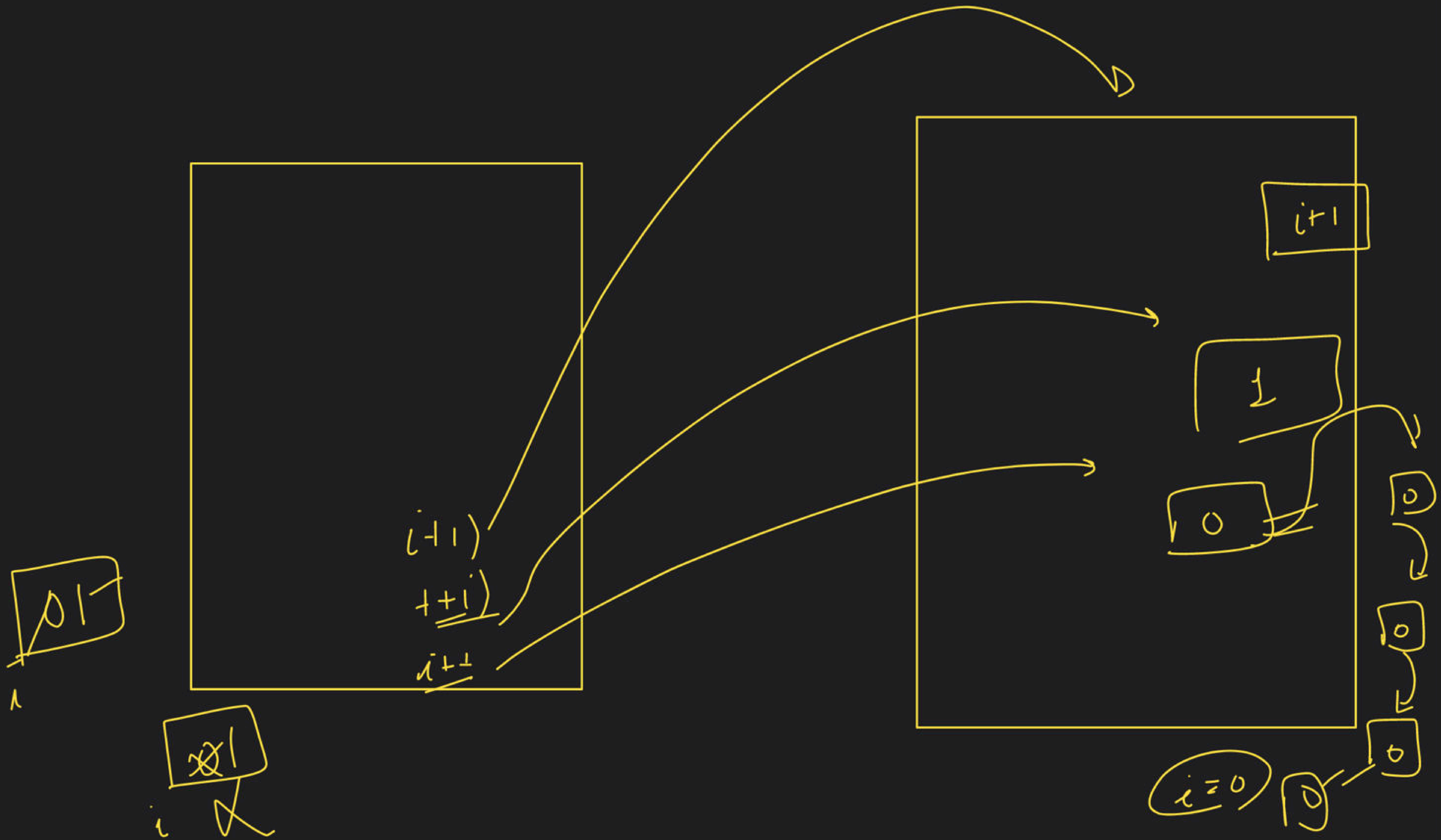
arr
n = 5
i = 0

$$\begin{tikzcd}
f(arr, n, \overset{i}{\underline{0}}) \arrow[d, "\text{print arr[0]}"'] \\
f(\underline{arr}, n, \underline{\underline{\overset{0+1=1}{i+1}}}) \arrow[d, "\text{print arr[1]}"'] \\
f(arr, n, \overset{1+1=2}{i+1}) \arrow[d, "\text{print arr[2]}"'] \\
f(arr, n, \overset{2+1=3}{i+1}) \arrow[d] \\
\cdots
\end{tikzcd}$$

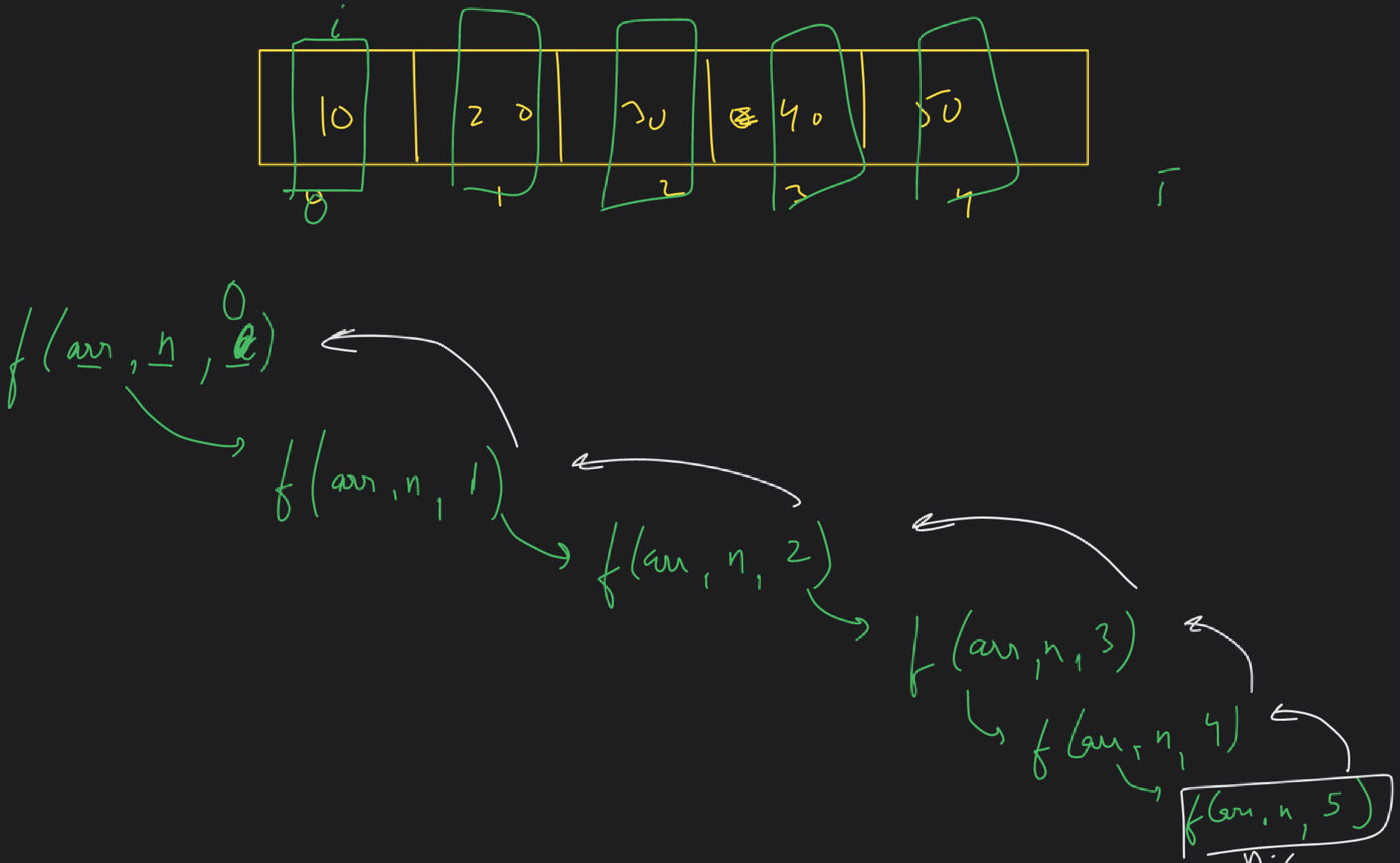


f(arr, n, 5) return

B.C



i+1
1
0
0
0
0
i=0
0
i+1
i+1
i+1
01
i
01
i


i
10
20
30
40
50
0
1
2
3
4
f(arr, n, 0)
f(arr, n, 1)
f(arr, n, 2)
f(arr, n, 3)
f(arr, n, 4)
f(arr, n, 5)
B.C

Max

Recursoin

10	30	15	21	44	26	17
0	1	2	3	4	5	6

loop

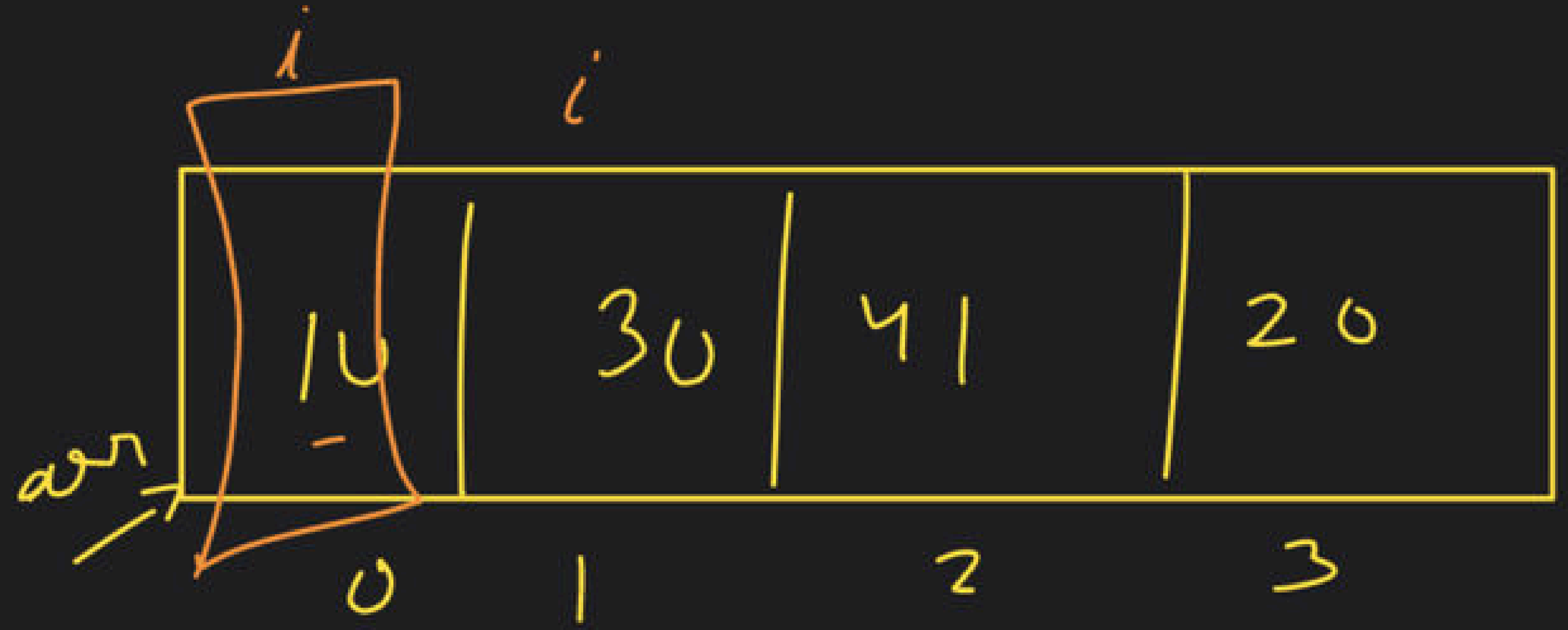
iteratin

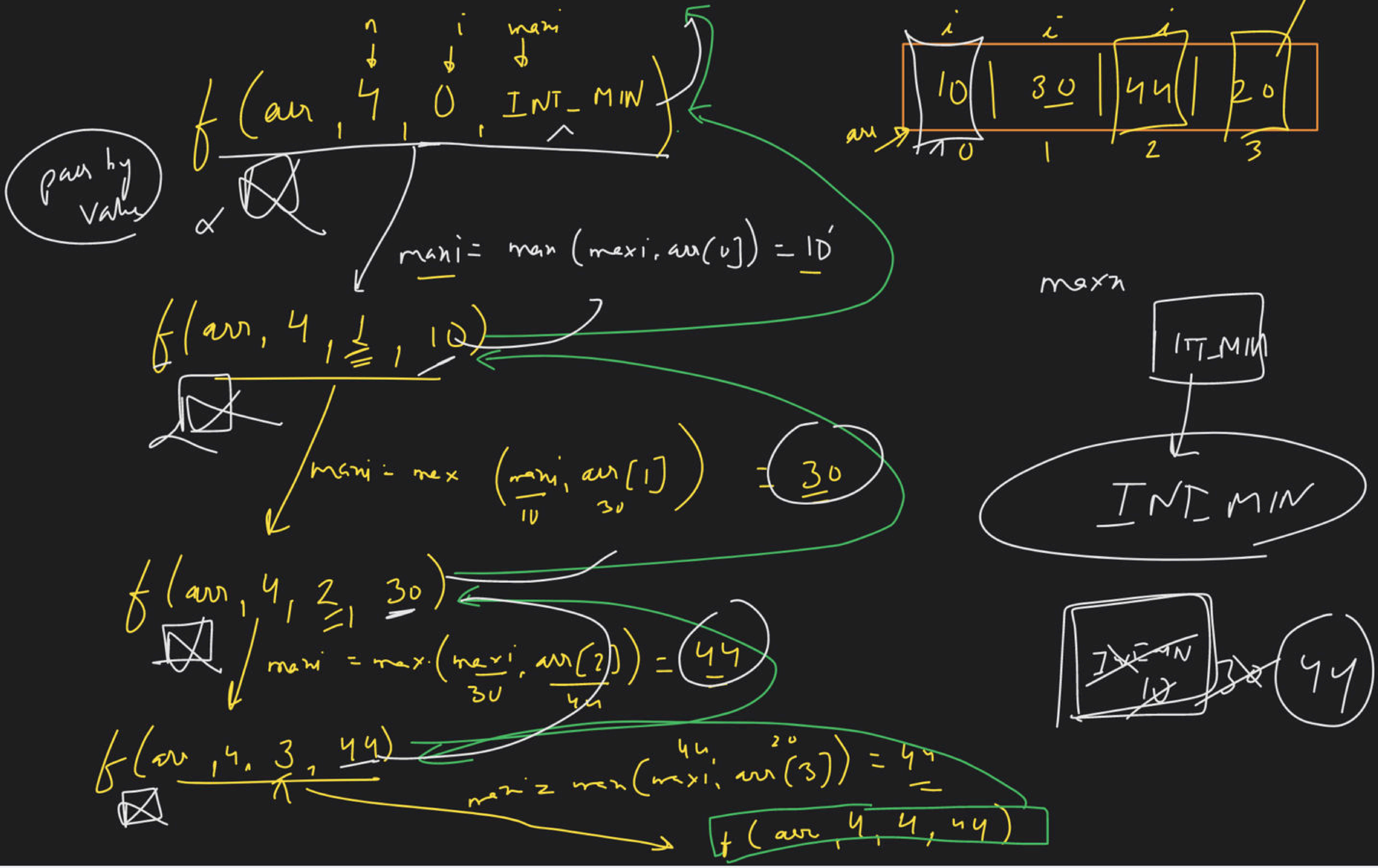
```
int maxi = INT_MIN;

for (int i=0; i<n; i++)
{
      if (arr[i] > maxi)
            maxi = arr[i];
}
```

Or

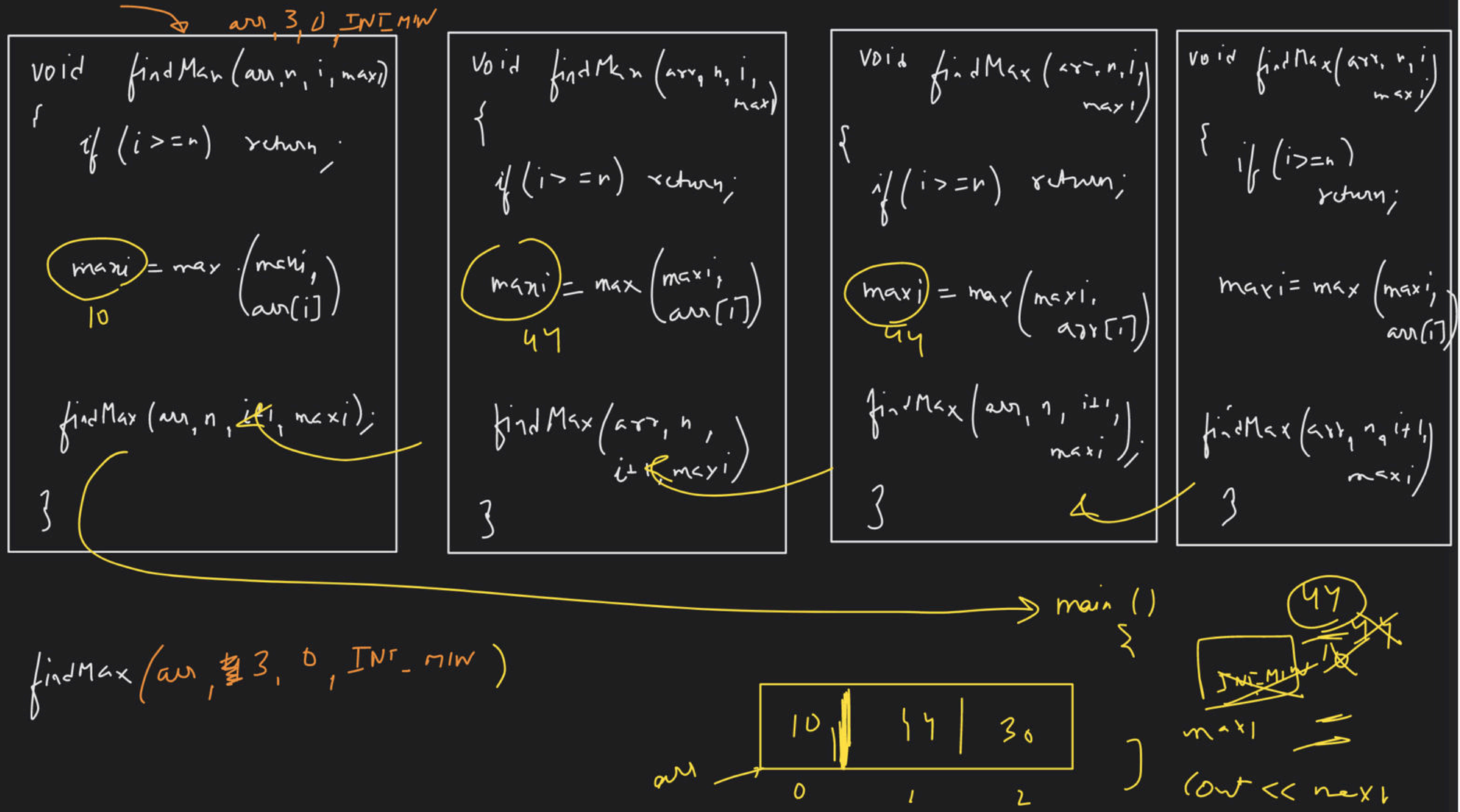
```
for (int i=0; i<n; i++)
{
    maxi = max(maxi, arr[i]);
}
```

$$\begin{tikzcd}
f(arr, \overset{n}{\downarrow}4, \overset{i}{\downarrow}0, \overset{maxi}{\downarrow}INT\_MIN) \arrow[d, "maxi\ update"] \\
f(arr, 4, 1, 10)
\end{tikzcd}$$





pass by value
n i maxi
f(arr, 4, 0, INT_MIN)
maxi = max(maxi, arr[0]) = 10
f(arr, 4, 1, 10)
maxi = max(maxi, arr[1]) = 30
10 30
f(arr, 4, 2, 30)
maxi = max(maxi, arr[2]) = 44
30 44
f(arr, 4, 3, 44)
maxi = max(maxi, arr[3]) = 44
44 20
f(arr, 4, 4, 44)
i i i
10 30 44 20
arr
0 1 2 3
maxn
INT_MIN
INT_MIN
~~INT_MIN~~
~~10~~
~~30~~
44



arr, 3, 0, INT_MIN

```
void findMax(arr, n, i, maxi)
{
    if (i >= n) return;

    maxi = max(maxi,
               arr[i])
    10

    findMax(arr, n, i+1, maxi);
}
```

```
void findMax(arr, n, i,
                maxi)
{
    if (i >= n) return;

    maxi = max(maxi,
               arr[i])
    44

    findMax(arr, n,
            i+1, maxi)
}
```

```
void findMax(arr, n, i,
                maxi)
{
    if (i >= n) return;

    maxi = max(maxi,
               arr[i])
    44

    findMax(arr, n, i+1,
            maxi);
}
```

```
void findMax(arr, n, i,
                maxi)
{
    if (i >= n)
        return;

    maxi = max(maxi,
               arr[i])

    findMax(arr, n, i+1,
            maxi)
}
```

findMax(arr, 3, 0, INT_MIN)

```
main()
{
    arr → | 10 | 44 | 30 |
            0    1    2

    maxi

    cout << maxi
}
```

44

~~INT_MIN~~ ~~10~~ ~~44~~

v –

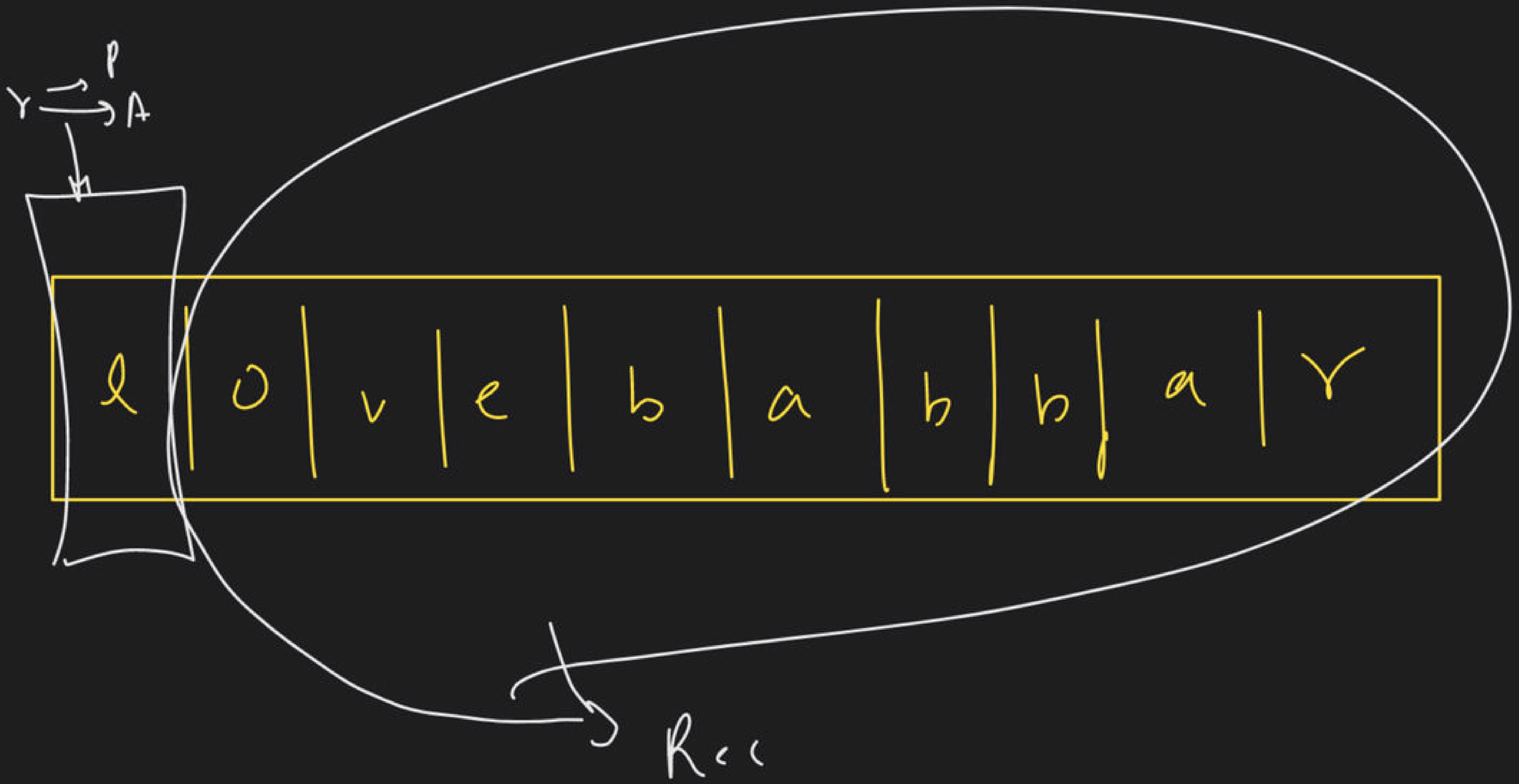
Rec

i/p → string str = "$\boxed{l}$ovebabbar"

Key → 'r'

→ r is present in str or not

(using recursion)


P
Y
A
l
o
v
e
b
a
b
b
a
r
Rec

Rec → array traverse ⟷ reverse traversal
- → by reference
- → void type function
- → non-void type function
- → by-reference → vector → store

find
- → check occurrence → T/F
- → index of occurrence → $\boxed{\text{int}}$
- → find all occurrence → $\boxed{\text{vector}}$

int* p = &a

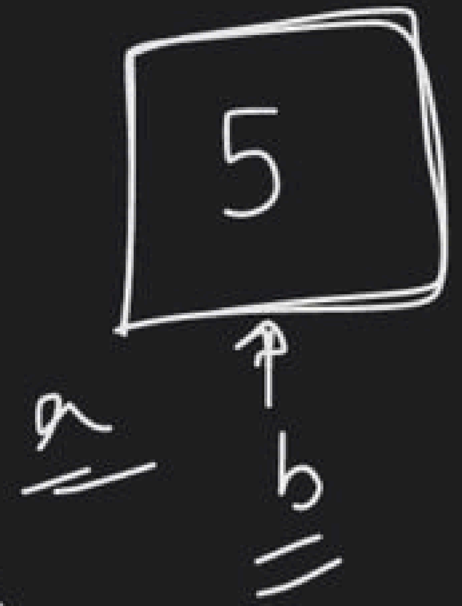
address of a

int a = 5

int & b = a;

reference variable

same memory different name

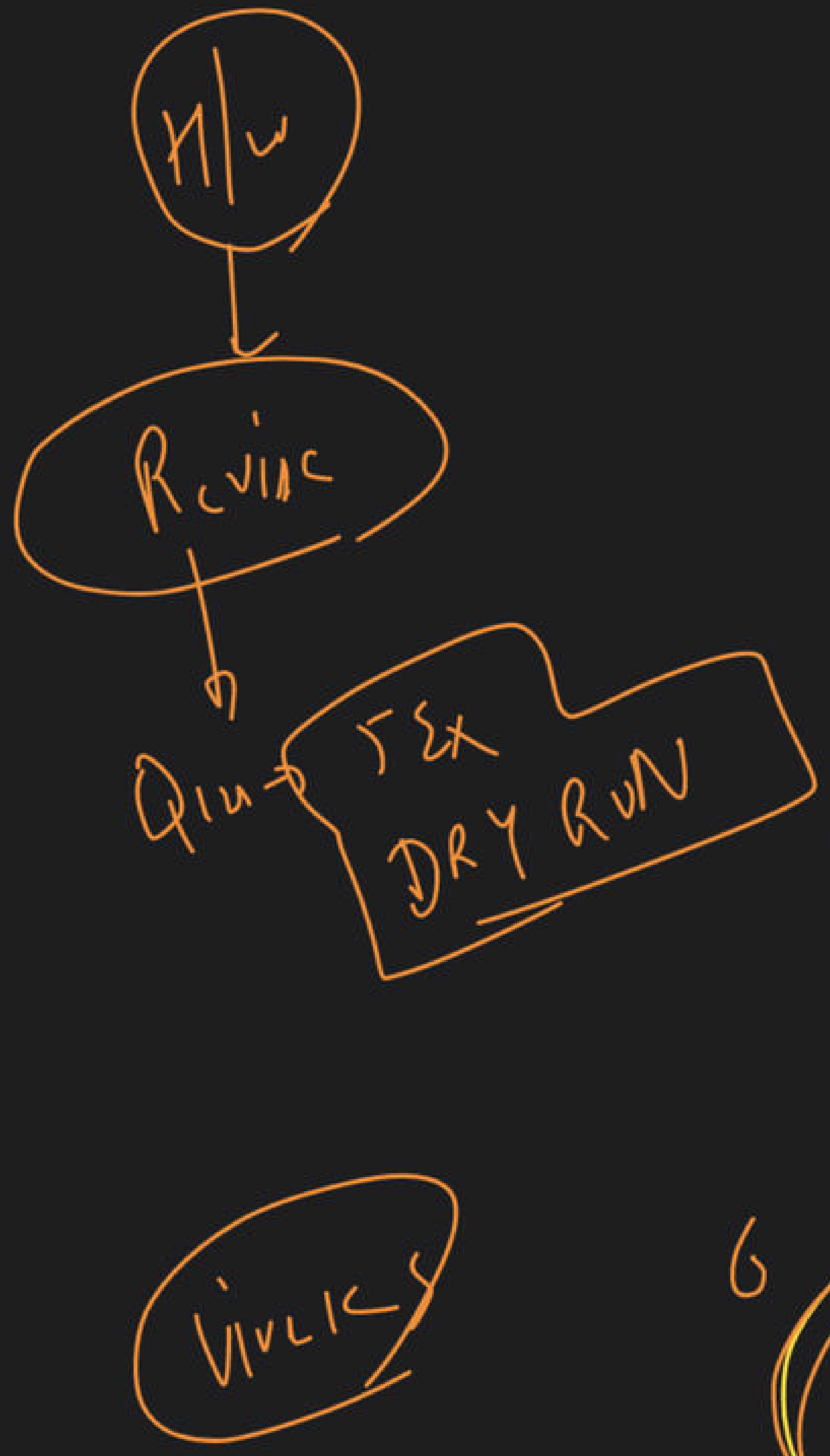


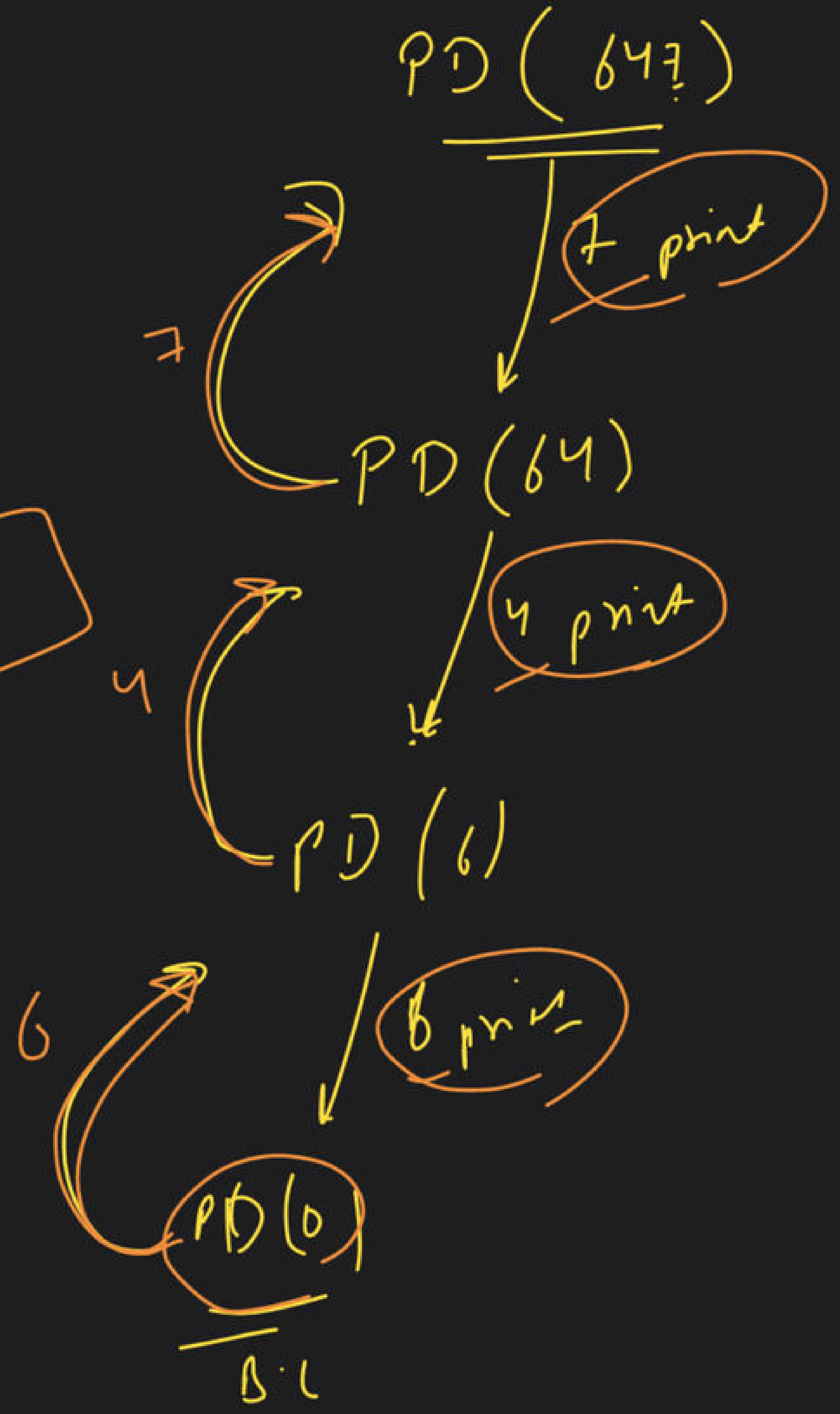
P.S

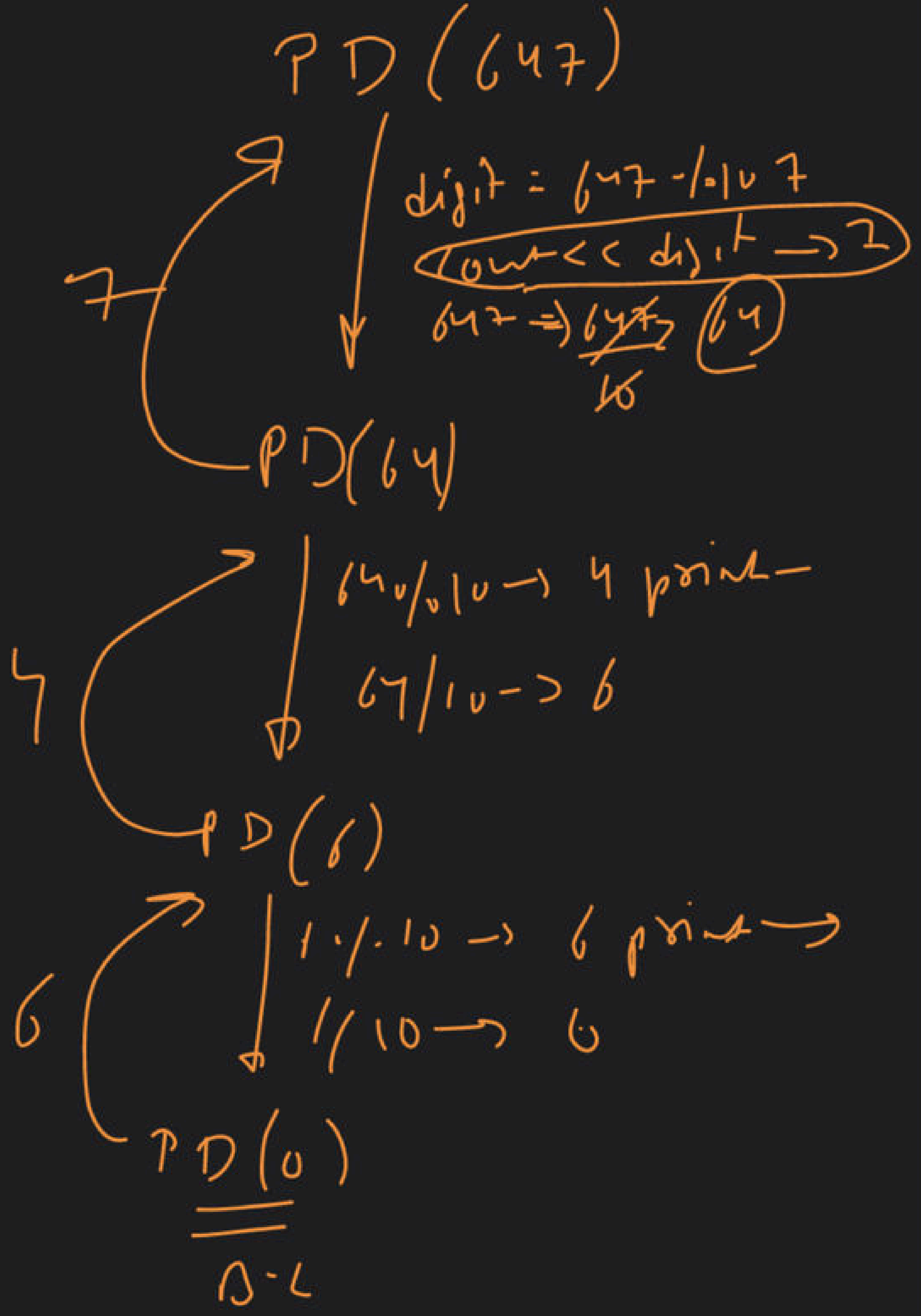
i/p → 647

o/p → print all digit of this no

$$\begin{tikzcd}
647 \arrow[r, "\%10"] \arrow[d, "/10"'] & 7 \\
64 \arrow[r, "\%10"] \arrow[d, "/10"'] & 4 \\
6 \arrow[r, "\%10"] \arrow[d, "/10"'] & 6 \\
0 \arrow[r] & \text{Return the}
\end{tikzcd}$$


H/W
Revise
Qn
TEX
DRY RUN
Viveks


PD (647)
7 print
7
PD (64)
4 print
4
PD (6)
6 print
6
PD(0)
B.C


PD (647)
digit = 647 % 10 7
cout << digit → 7
647 → 647/10 → 64
7
PD(64)
64%10 → 4 print
64/10 → 6
4
PD (6)
6%10 → 6 print
6/10 → 0
6
PD (0)
B.C