

≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course

Multiple Recursion calls

To exit full screen, press **Esc**



TUF



0:02 / 16:44 • Introduction >



Multiple Recursion Calls



TUF

Multiple Recursion Calls



TUF

Multiple Recursion Calls

$f()$
 {

$f()$
 {

$f()$
 {

$f()$ }
 $f()$ }
 } 2 times



Multiple Recursion Calls

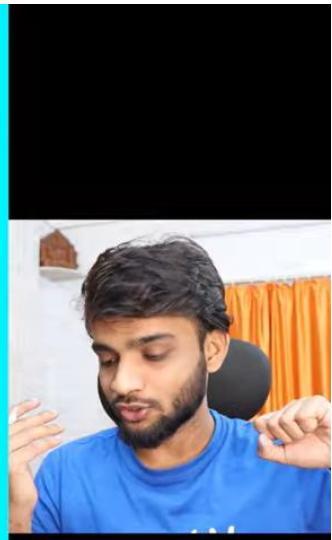
$f()$
}

$f()$
}

$f()$
}

$f()$
 $f()$
}
}

2 times



Multiple Recursion Calls

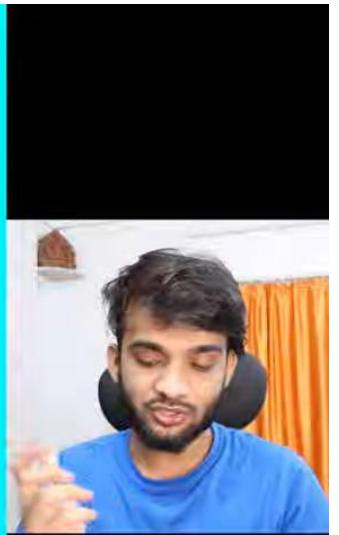
$f()$
 {

$f()$
 {

$f()$
 {

$f()$ }
 $f()$ }
 } 2 times





TUF

Fibonacci

0 1 1 2



TUF

Fibonacci

0 1 1 2 3 5



TUF

Fibonacci

0 1 1 2 3 5 8 13 21 34 ..



TUF

Fibonacci

0 1 1 2 3 5 8 13 21 34 ...
↑ ↓ ↑ ↓ =
0th 1st 2nd 3rd - - -



Re 5. Fibonacci | Problems | Strivers A2Z DSA Course



0 1 1 2 3 5 8 13 21 34 ...
↑ ↓ ↑ ↓ ↑ - - -
 0^{th} 1st 2nd 3rd



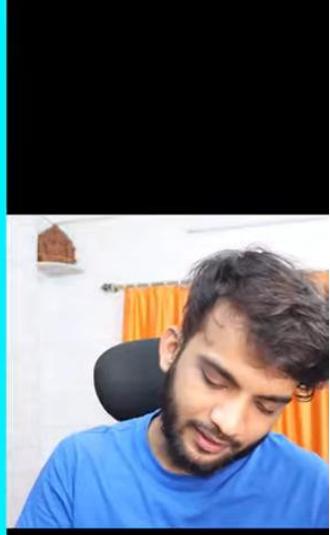
N. → $f(N)$ → N^{th} fib.



2:11 / 16:44 • Fibonacci Number >



TUF



TUF

$$\left. \begin{array}{l} f[0] = 0 \quad f[1] = 1 \\ f[n] \quad (i=2 \rightarrow n) \\ f[i] = f[i-1] + f[i-2] \end{array} \right\}$$



TUF

≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course


$$\begin{cases} f(n) \\ \hat{f}(n \leq 1) \end{cases}$$


TUF

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

$f(n)$

{
 if ($n \leq 1$)

 return n;

 return $f(n-1) + f(n-2)$;

}



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

```
f(n)
{
    if (n <= 1)
        return n;
    return f(n-1) + f(n-2);
}
```



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

```
f(n)
{
    if (n <= 1)
        return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
```



```
f(n)
{
    if (n <= 1)
        return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}

main()
{
    n ← 4
    print(f(n))
}
```



```
f(n) {
    if (n <= 1) {
        return n;
    }
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
main() {
    n ← 4
    f(1, f(n))
}
```



```
f(n) {
    if (n <= 1) return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
main() {
    n ← 4
    f(1, f(n))
```



```
f(4)
{
    if (n <= 1) x
        return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
main()
{
    n ← 4
    f(1)(n)
}
```



```
f(n) {
    if (n <= 1) return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
main() {
    n ← 4
    f(1)(n)
}
```



```

    }  

    f(1)  

    {  

        if(n <= 1) x  

        return n;  

        last = f(n-1)  

        sLast = f(n-2)  

        return last + sLast;  

    }
    main()
    {
        n ← 4
        ↪ f(4)
    }

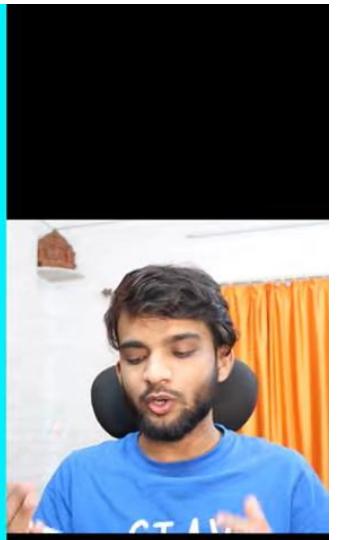
```

```

f(3)
{
    if() x
    last = f(2)
    sLast = f(1)
    return last + sLast
}

```





```
g(y)
{
    if (n <= 1) x
        return n;
    last = f(n-1)
    sLast = f(n-2)
    return last + sLast;
}
main()
{
    n ← y
    ↪ f(y)
}
```

Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



```
g(?)  
{  
    if (n <= 1) x  
    return n;  
  
(o) last = g(n-1)  
sLast = g(n-2)  
  
return last + sLast;
```

```
g(?)  
{  
    if ( ) x  
    last = g(?)  
    sLast = g(?)  
    return last + sLast  
}
```

TUF

$f(3)$

$y()$ *

$$\text{last} = f(2)$$

$$\text{slast} = f(1)$$

return last + slast

}

-slast;

$f(2)$

y *

last

.



TUF



X

$f(2)$

$\{ \hat{y} X$

$rest = f(1)$

$slant = f(0)$

then $rest + slant$

$m + s \text{ less } ?$

}

~~X~~

$f^{(2)}$

$\{ \hat{y} \}$

$l_{out} = f^{(1)}$

$s_{out} = f^{(0)}$

$return \ l_{out} + s_{out}$

}

$f^{(1)}$

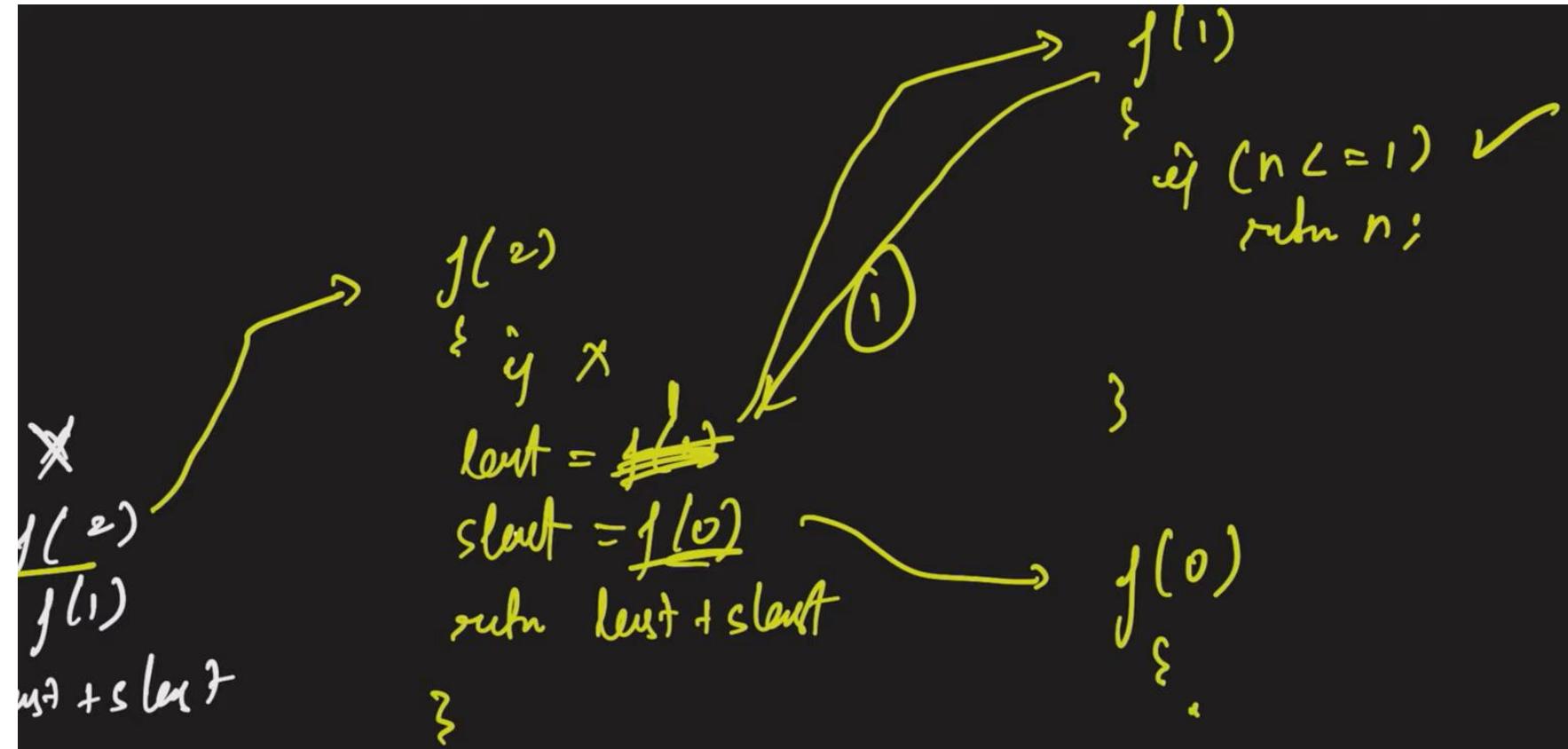
$\{ \hat{y} \}$

$\hat{y} (n <= 1) \checkmark$

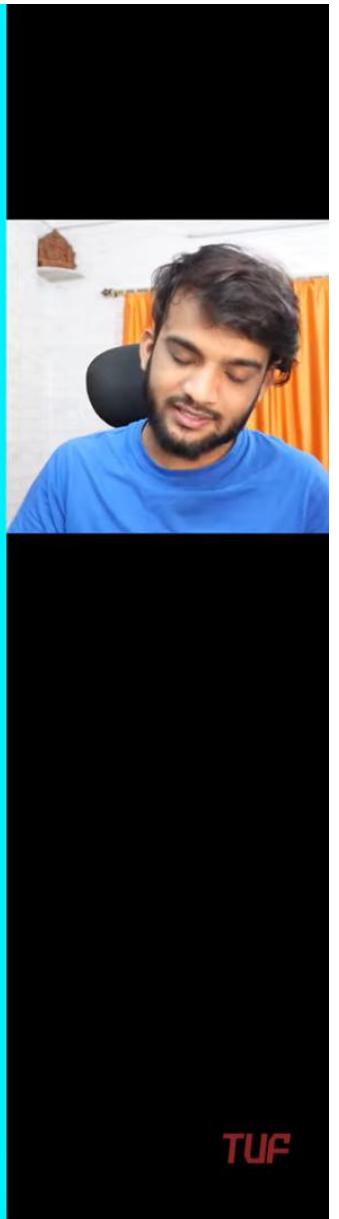
$return \ n;$

}





TUF



~~X~~

$f(2)$

$\{ \hat{y} \}$

$\text{least} = \cancel{\dots}$

$\text{slowt} = \cancel{\dots}$

sum least + slowt

$\}$

$f(1)$

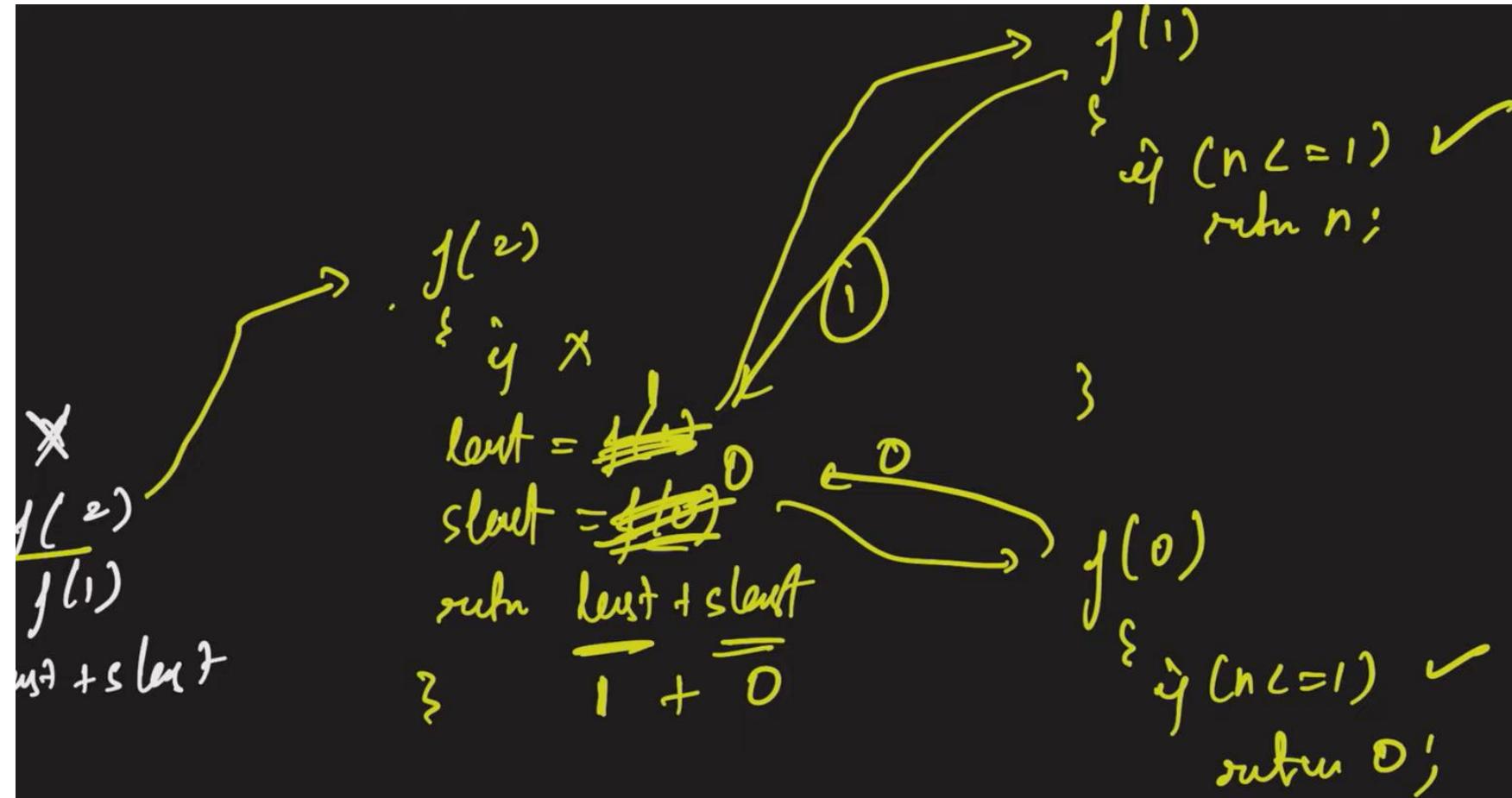
$\{ \hat{y} (n <= 1) \checkmark$

$\text{sum } n;$

$f(0)$

$\{ \hat{y} (n <= 1) \checkmark$

$\text{sum } D;$



y

$\hat{y}(n \leq 1) \times$
return $n;$

$last = f(n-1)$

$slast = f(n-2)$

return $last + slast;$

$am()$

$n \leftarrow 9$

$f(3)$

$\hat{y}() \times$

$last = \underline{\underline{f(1)}}$

$slast = f(1)$

return $last + slast$

}

$f(2)$

$\hat{y} \times$

$last = \cancel{\cancel{1}}$

$slast = \cancel{\cancel{0}}$

return $last + slast$

 }

TUF

$f(3)$

$\hat{y}() \times 1$

$last = f(2)$

$start = f(1)$

sum $last + start$

}

+ start;

$f(2)$

$\hat{y} \times$

$last = \cancel{f(1)}$

$start = \cancel{f(0)}$

sum $last + start$

$\overline{1} + \overline{0}$

$f(1)$

$\hat{y}(n=1)$

sum 1



$f(3)$

$i(j)$

$last = \underline{\underline{j}}$

$start = \underline{\underline{j+1}}$

$sub last + start$

}

+ start;

$f(2)$

$i(j)$

$last = \underline{\underline{j}}$

$start = \underline{\underline{j+1}}$

$sub last + start$

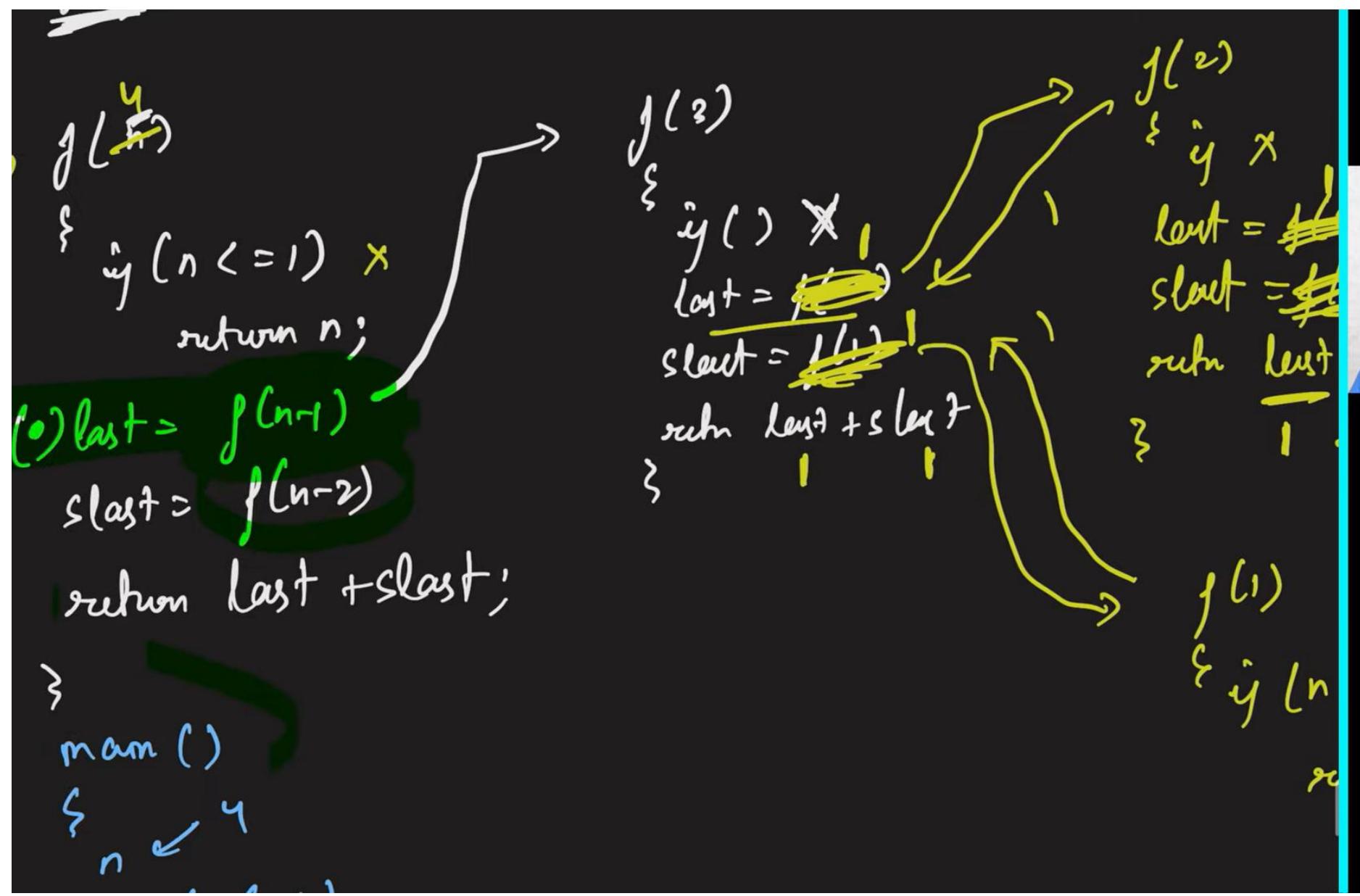
$\overline{1} + \overline{0}$

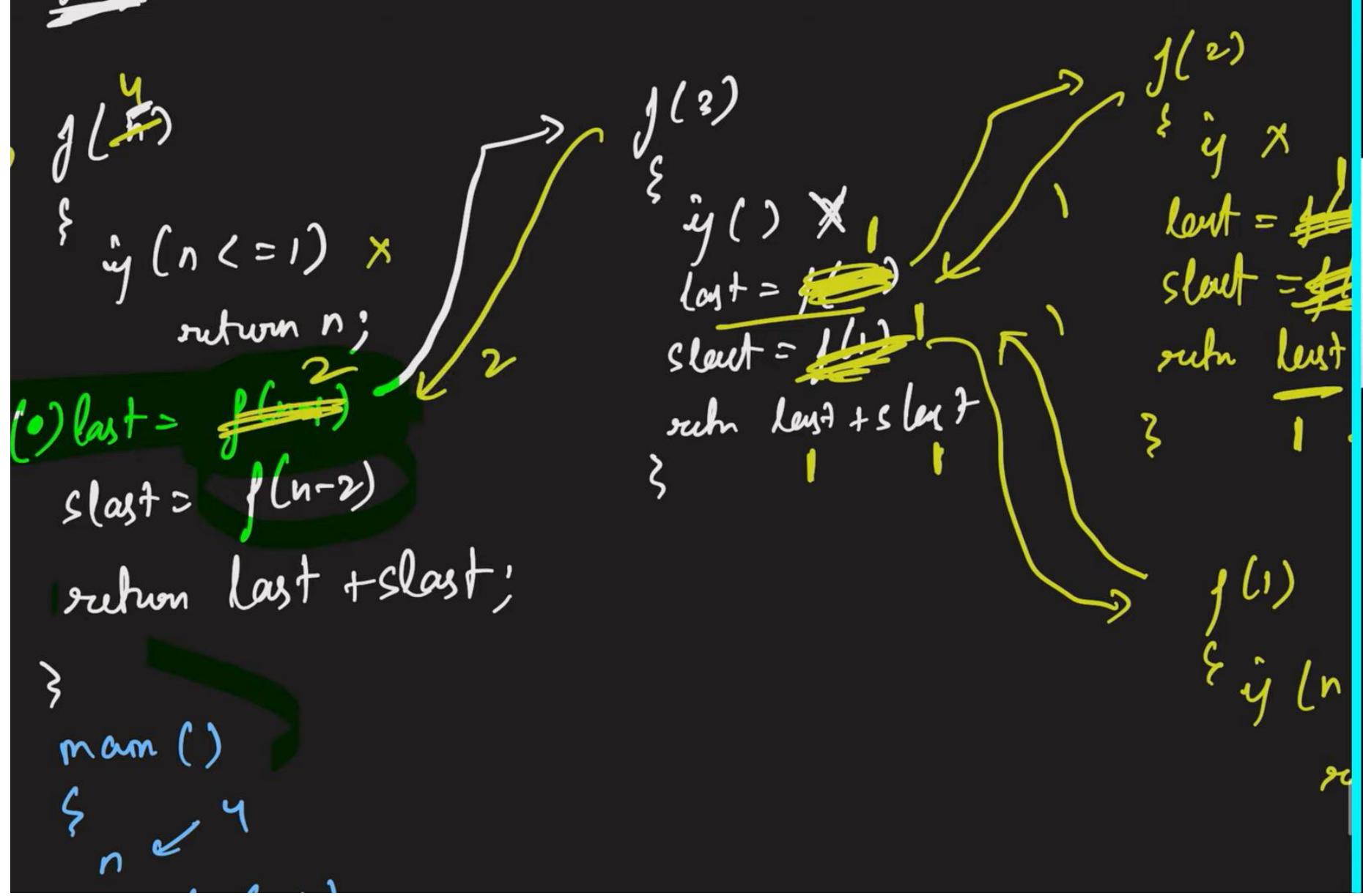
$f(1)$

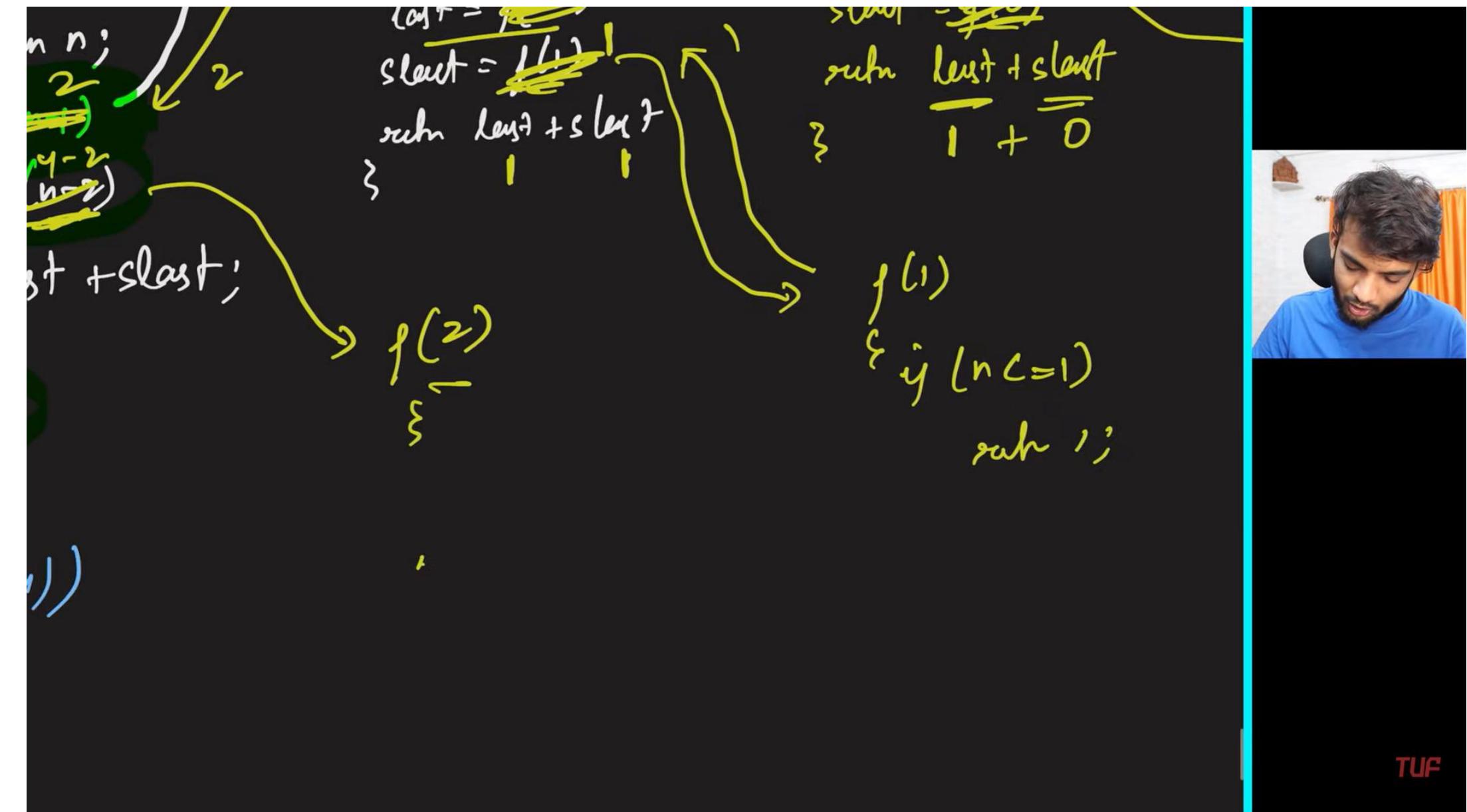
$i(j) \quad n=1$

$sub 1;$









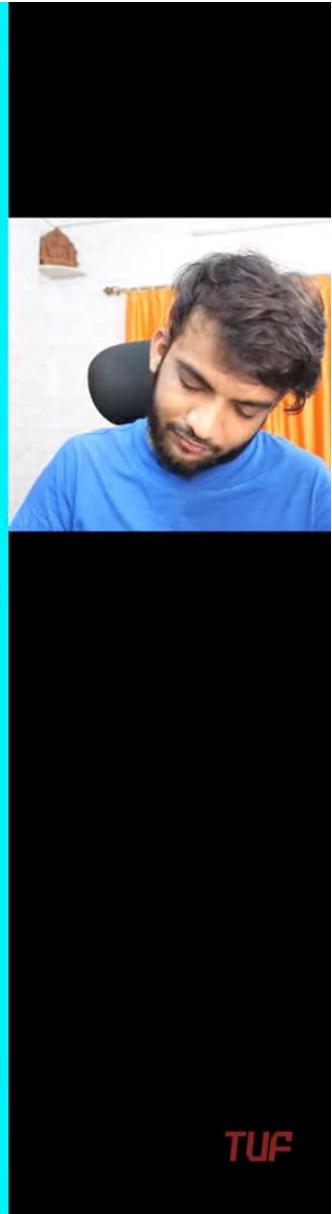
$$\text{start} = \cancel{\text{last}} - \cancel{\text{last}}$$
$$\text{return } \text{last} + \text{start}$$
$$\frac{1}{1} + \frac{0}{0}$$

$$f(2)$$
$$\underbrace{\quad}_{\dots}$$

$$\text{last} = f(1)$$

$$\text{start} = f(0)$$

$$f(1)$$
$$\& i \leq n \& i = 1$$
$$\text{return } 1;$$





BT + stack_i

$f(2)$

$\{$

$y()$ ↗

$\text{left} = f(1)$

$\text{right} = f(0)$

$\text{sub } \underline{1+0}$

$f(1)$

$\{ y(n=1)$

$\text{sub } 1;$



TUF

```
return n;
)
last = f(1)
sLast = f(n-2)
return last + sLast;

main()
{
    n ← 4
    print(f(n))
}
```

return n;

last = f(1)

sLast = f(n-2)

return last + sLast;

f(2)

y()

last = f(1)

sLast = f(0)

return 1 + 0;

return last + sLast;

1 +

f(1)

& y(n =

return

```

        return n;
    }

    () last = f(2)
    sLast = f(n-2)

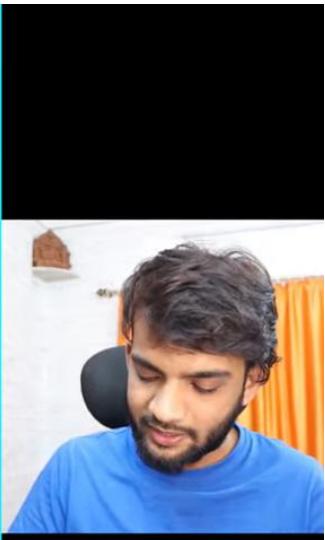
    return last + sLast;
}

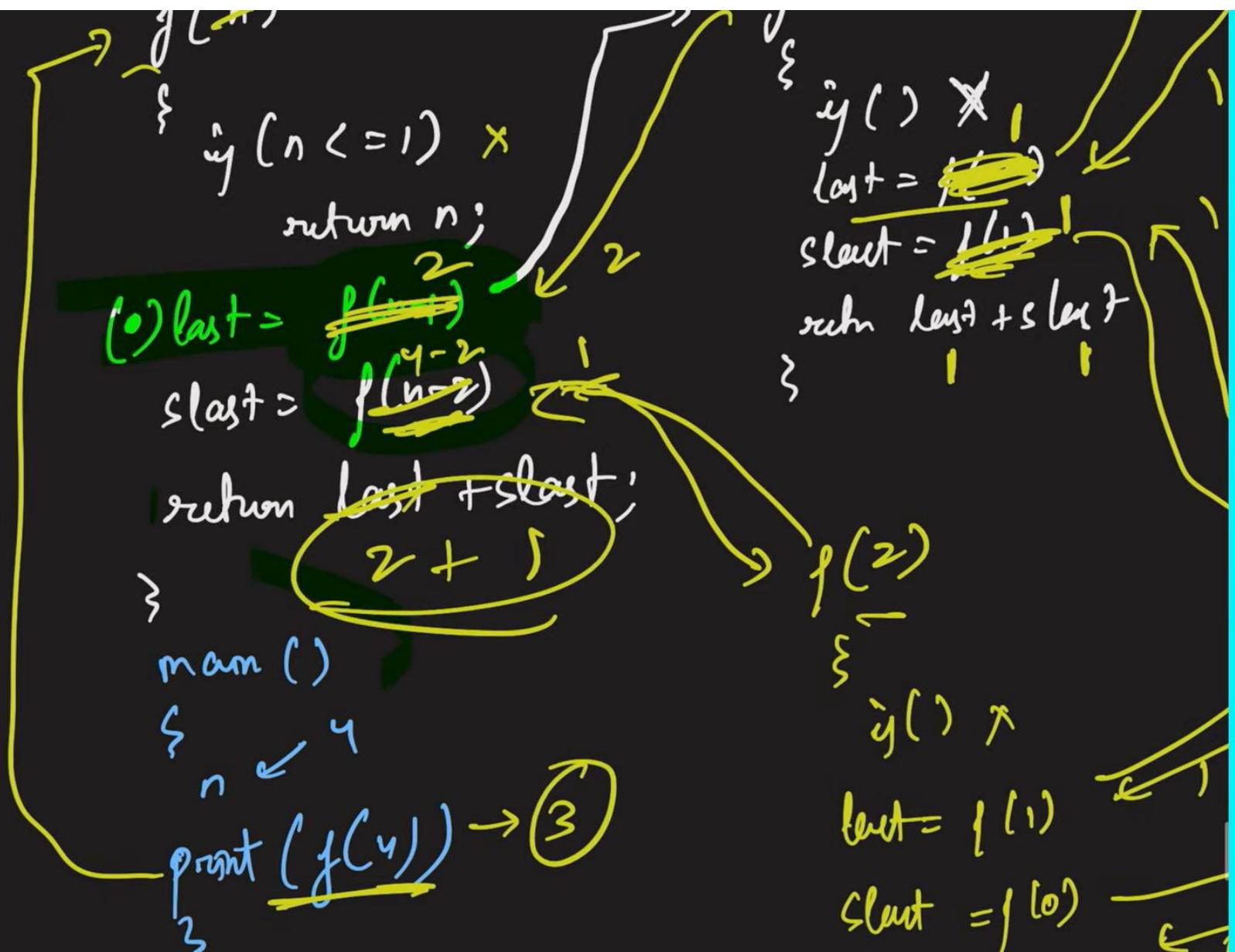
main()
{
    n ← 4
    print(f(4))
}

```

~~last = f(1)~~
~~sLast = f(0)~~
 return last + sLast

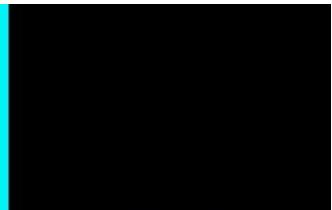
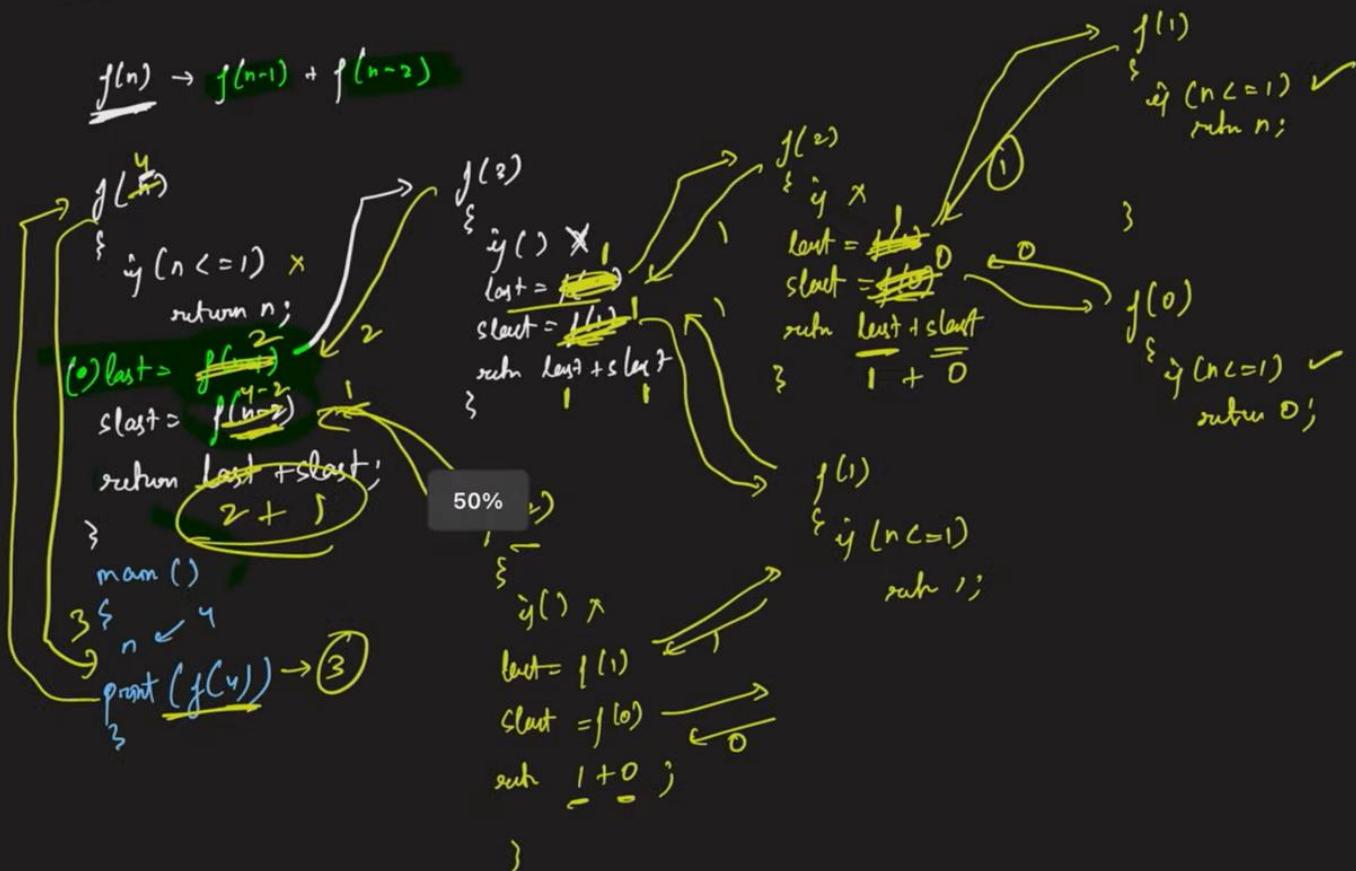
~~f(2)~~
~~f(1)~~
~~f(0)~~
 return 1 + 0 ;





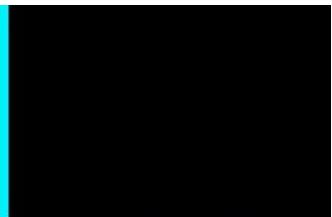
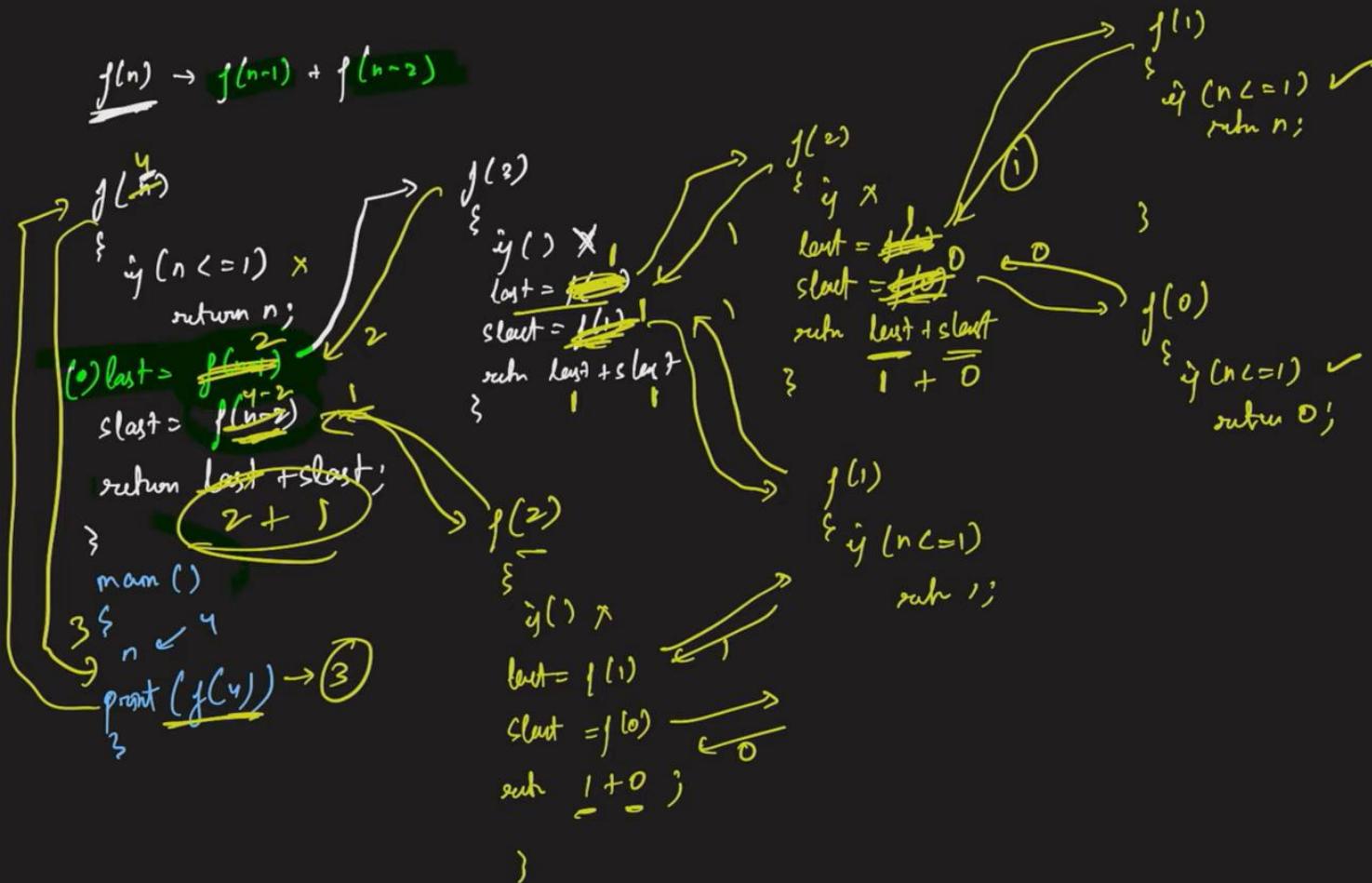
$$\left| \begin{array}{l} f^{(r)} = f^{(r)} \\ f^n(i=2 \rightarrow n) \\ f^S_i = f^S_{i-1} + f^S_{i-2} \end{array} \right|$$

$$f(n) \rightarrow f(n-1) + f(n-2)$$



$$\boxed{f^{\{i\}} = f^{\{i-1\}} + f^{\{i-2\}}}$$

$$f(n) \rightarrow f(n-1) + f(n-2)$$



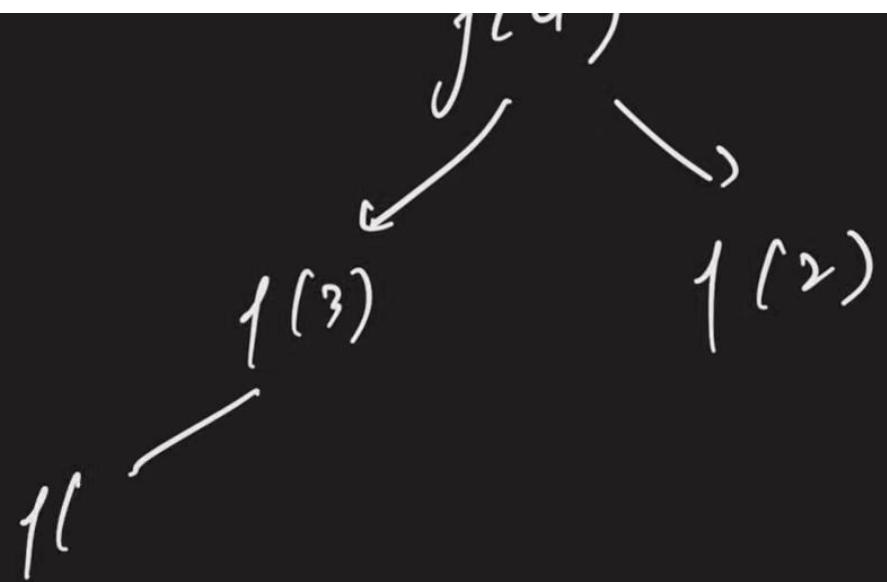
$f()$

{

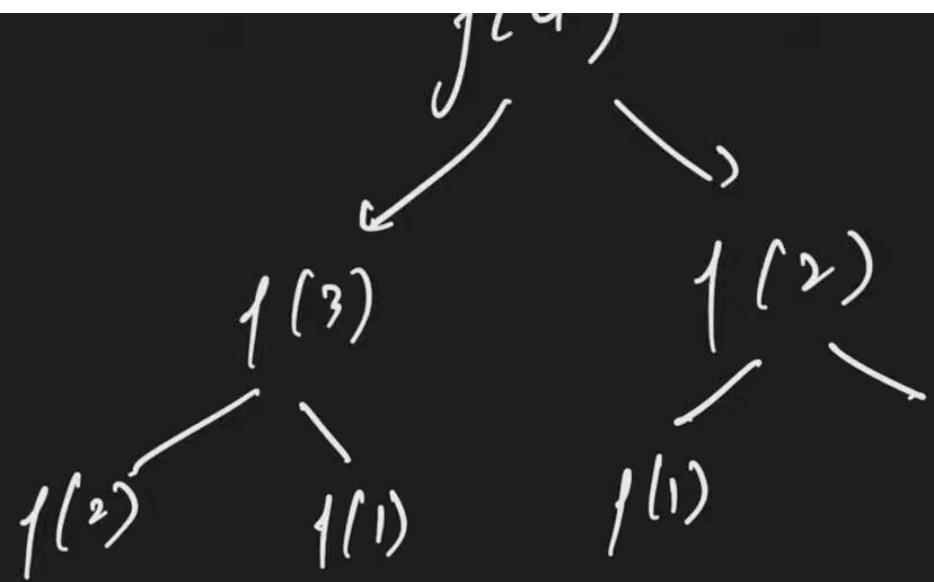
$f()$

$f()$

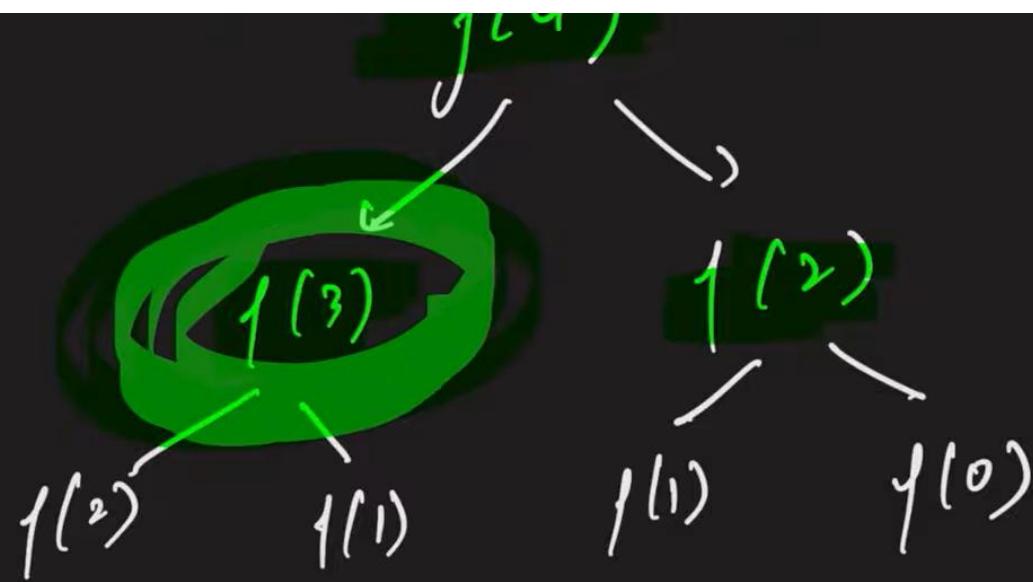
$f()$



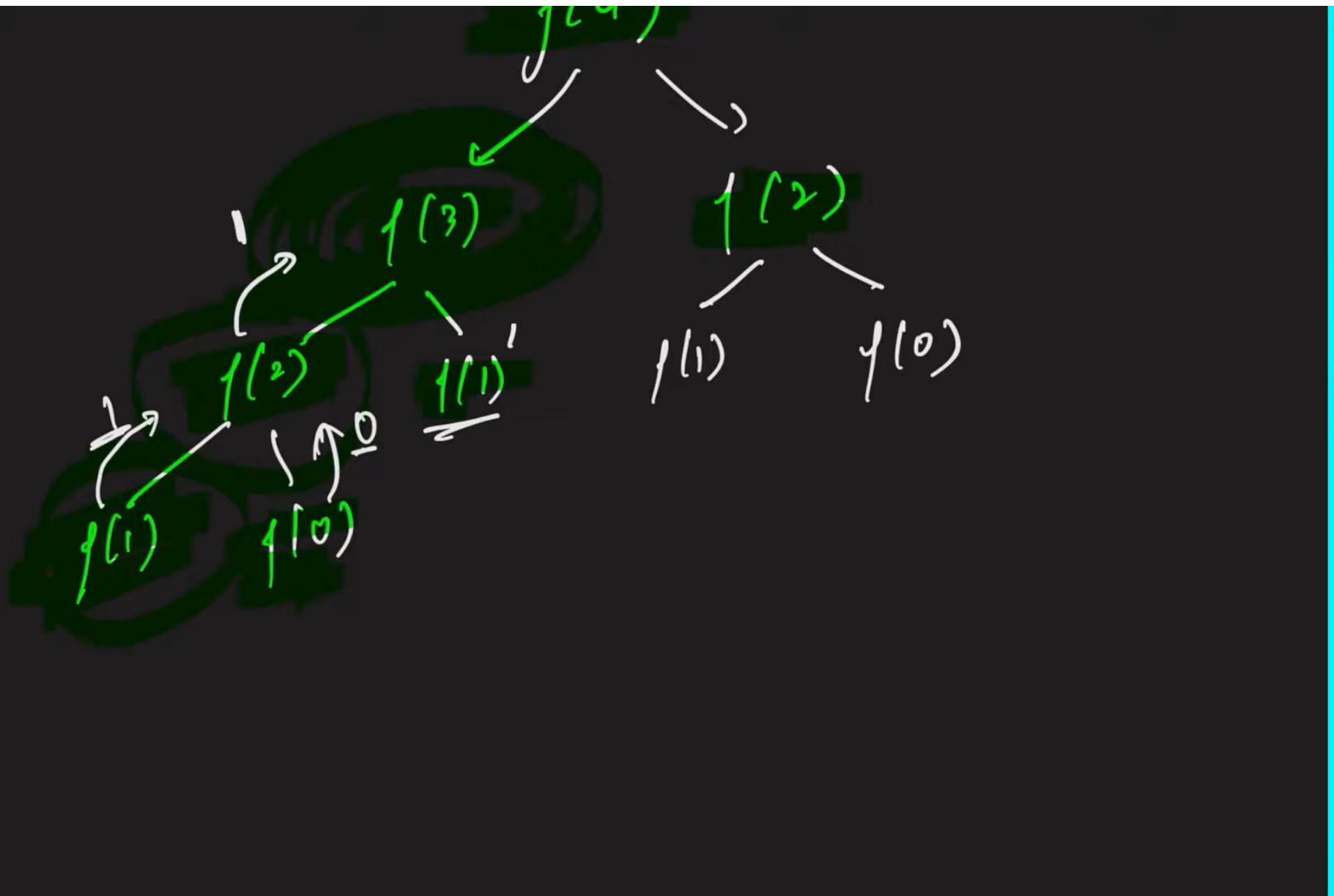
TUF



TUF



TUF

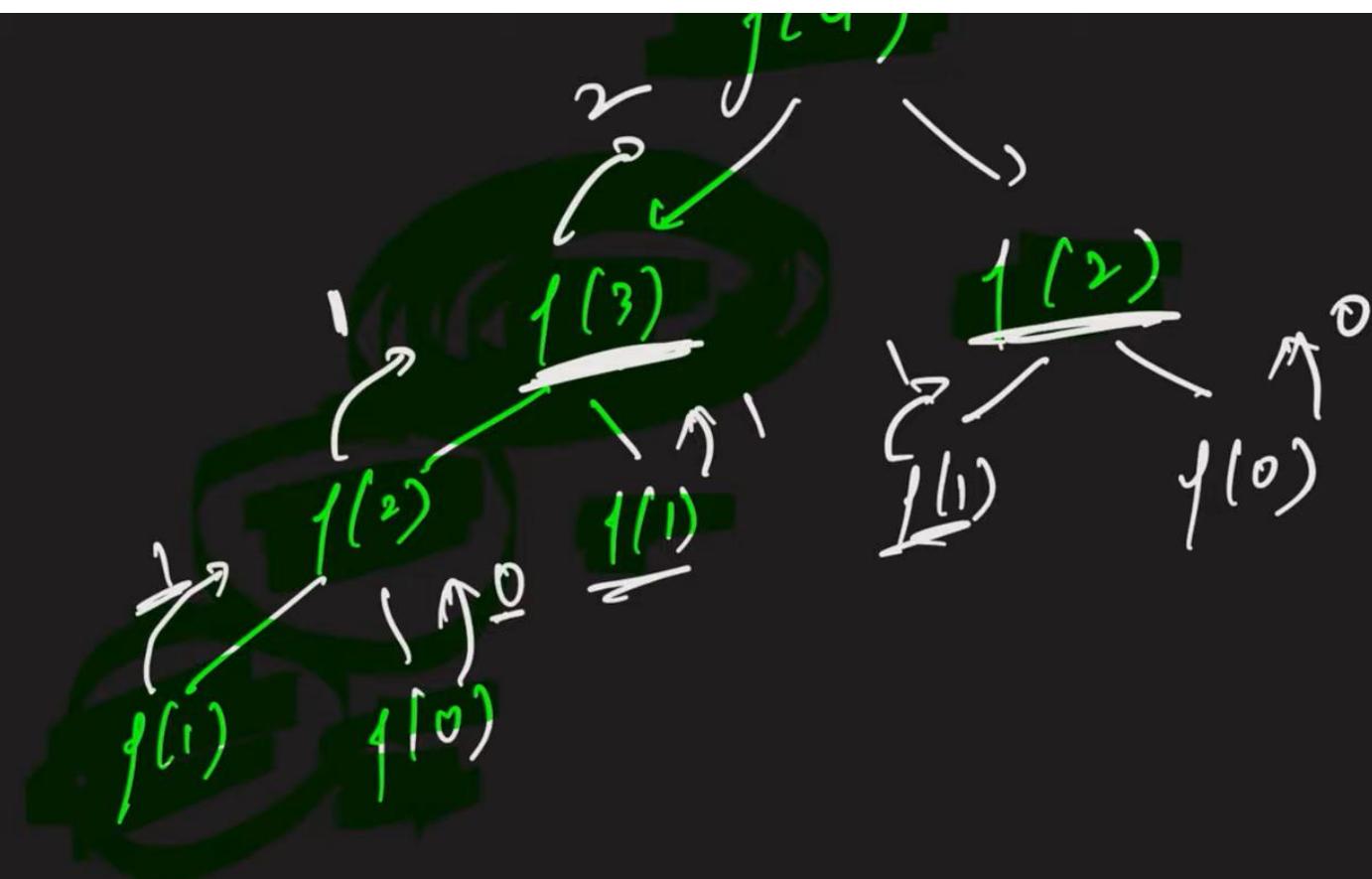


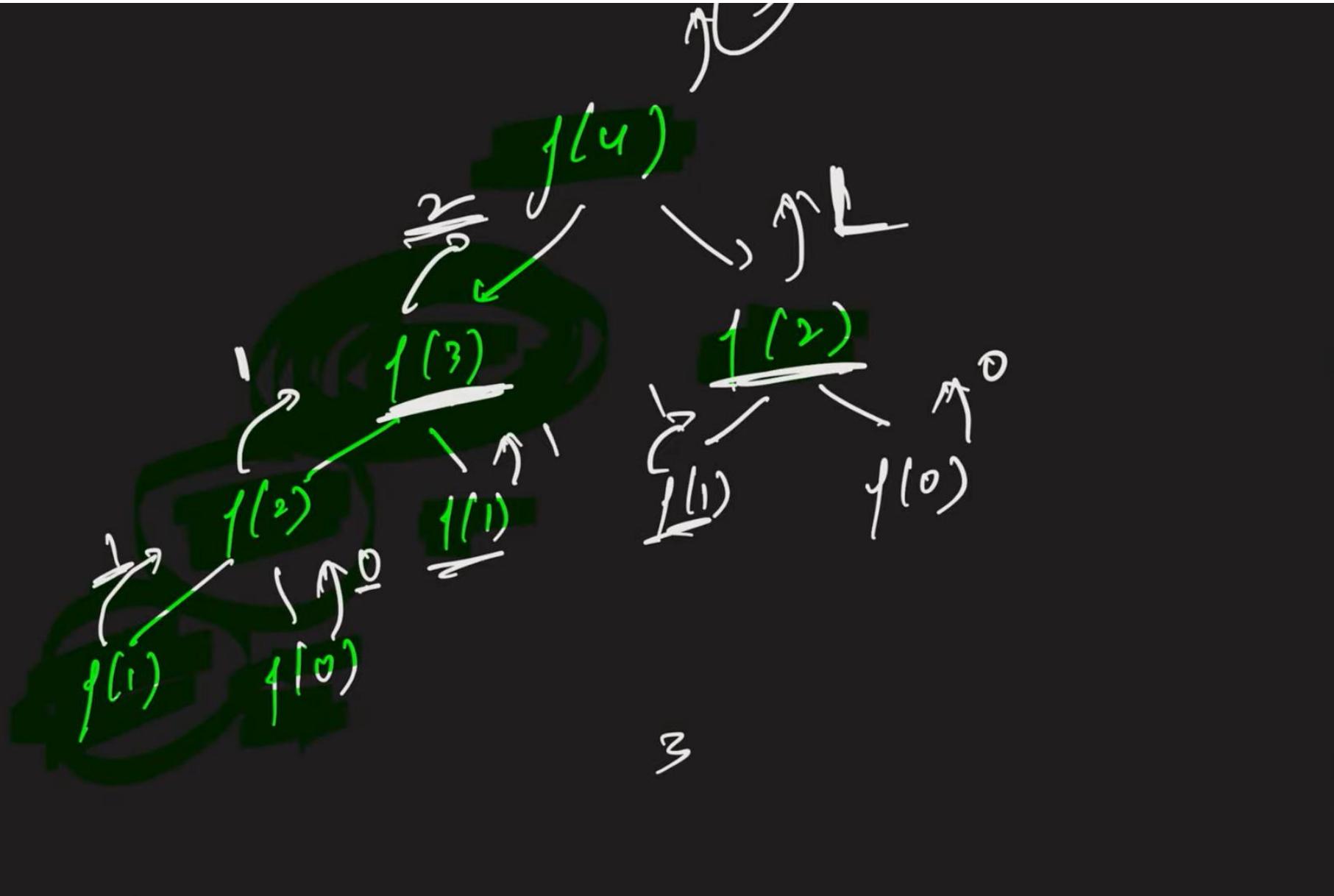
TUF

Let's give back to the community :) 

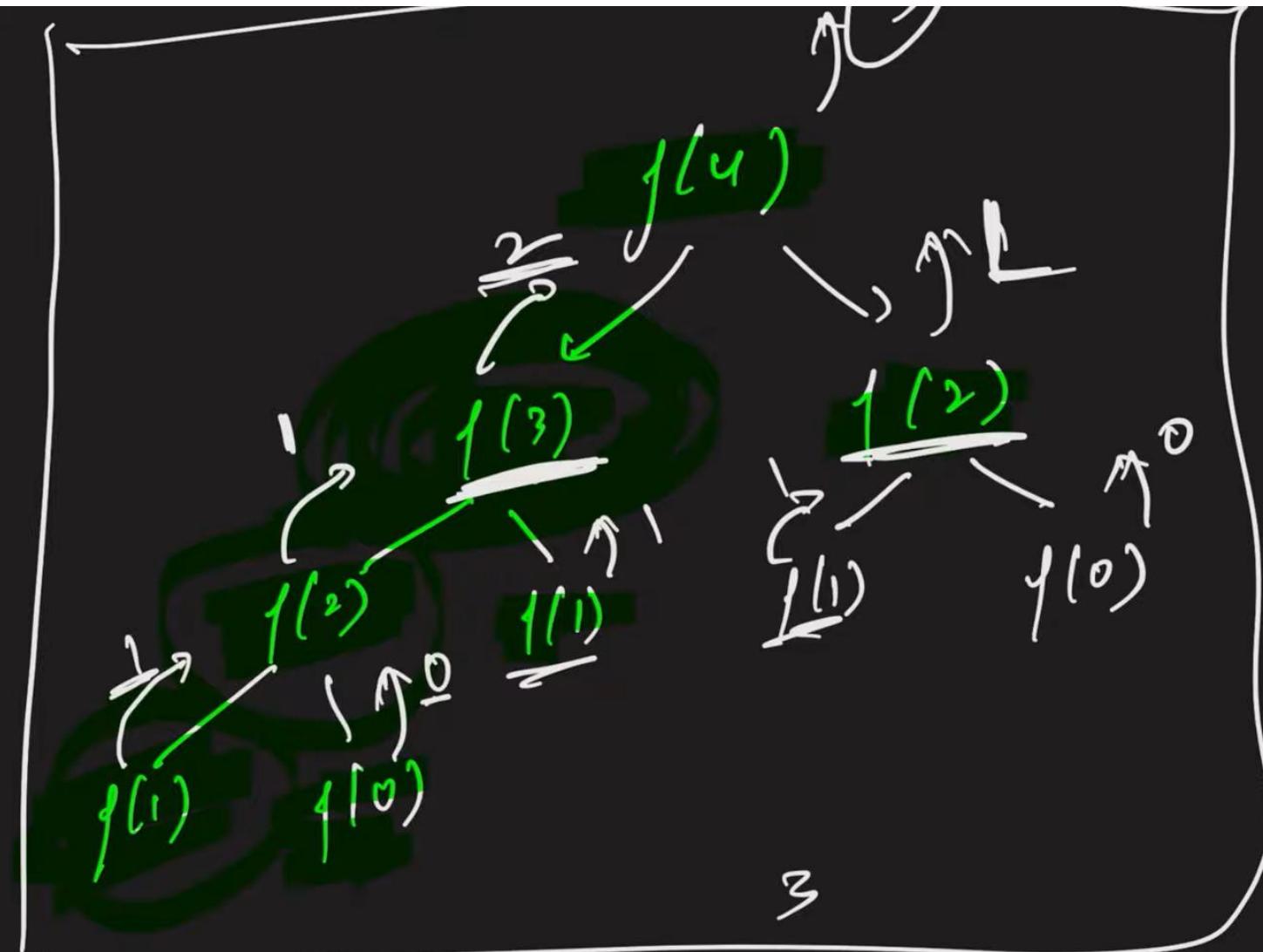


TUF





TUF



$\text{sub } \frac{1+0}{1-0} = j$

}

$j(3)$

$j(4)$

$j(3)$

$j(2)$

$j(1)$

$j(0)$

$j(L)$

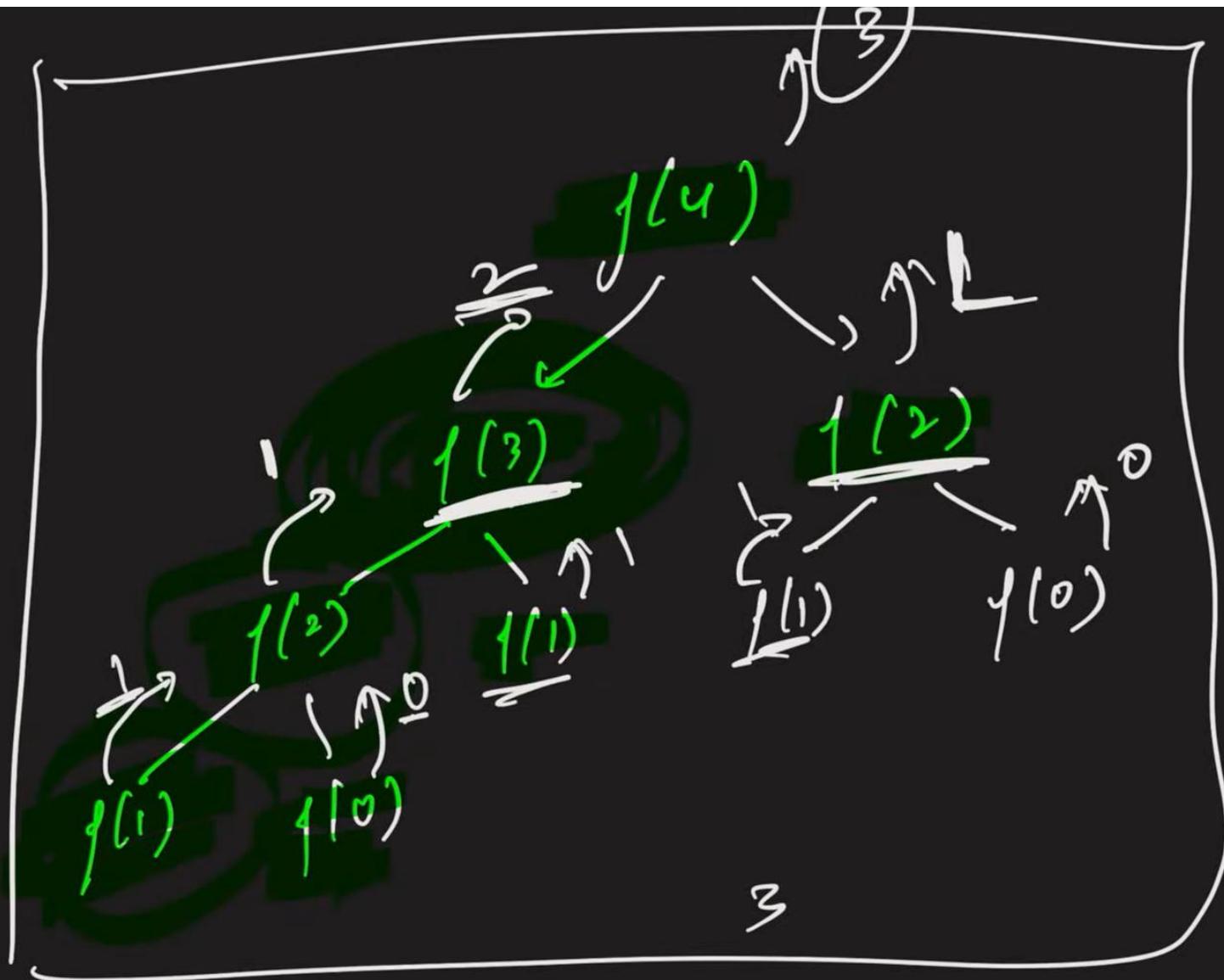
$j(2)$

$j(1)$

$j(0)$



TUF



TUF

≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course

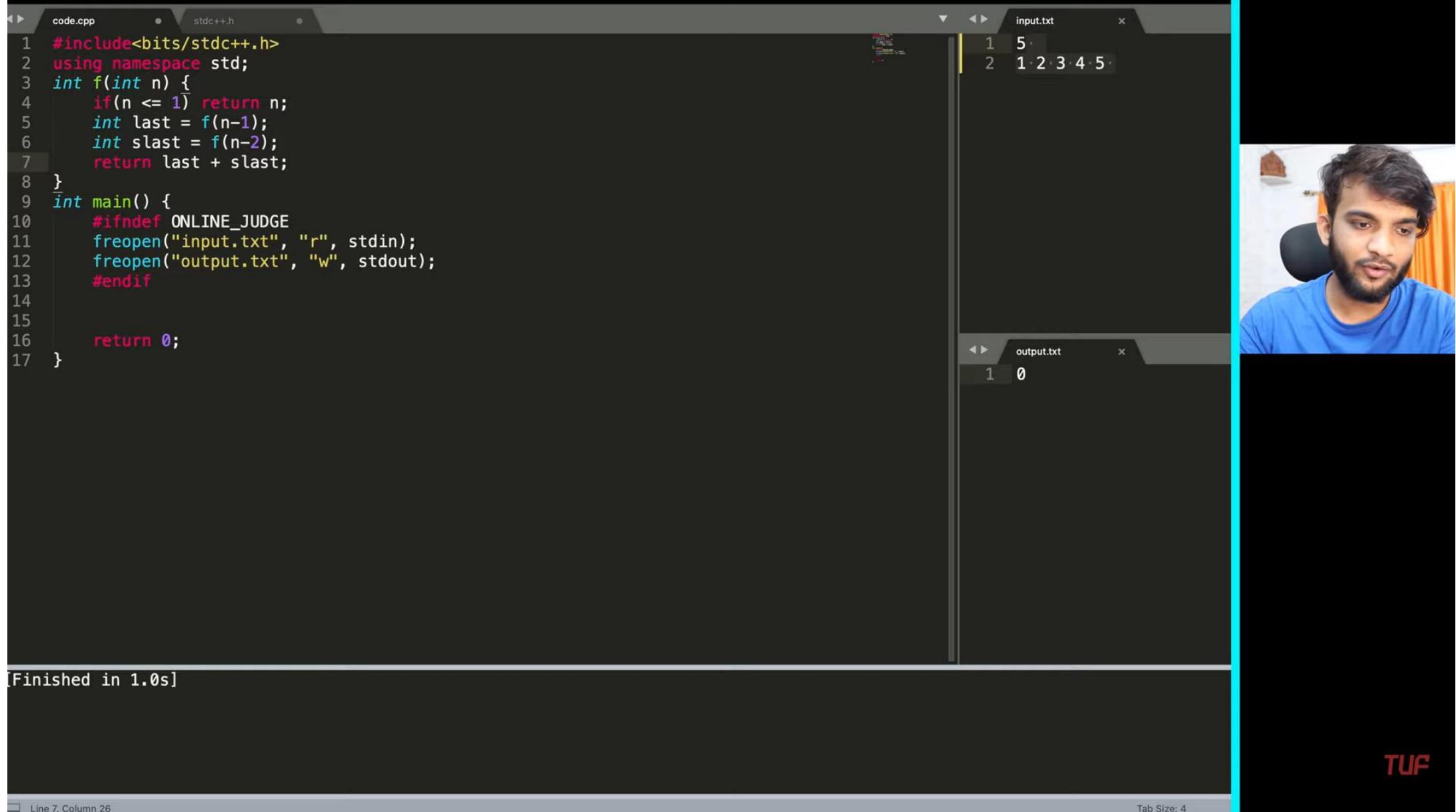


TUF

◀ ▶ 🔍 13:07 / 16:44 • Code >



CC HD ⌂



```
code.cpp stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 int f(int n) {
4     if(n <= 1) return n;
5     int last = f(n-1);
6     int slast = f(n-2);
7     return last + slast;
8 }
9 int main() {
10 #ifndef ONLINE_JUDGE
11     freopen("input.txt", "r", stdin);
12     freopen("output.txt", "w", stdout);
13 #endif
14
15
16     return 0;
17 }
```

[Finished in 1.0s]

input.txt

1 5
2 1 2 3 4 5

output.txt

1 0

TUF

code.cpp stdc++.h

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int f(int n) {
4     if(n <= 1) return n;
5     int last = f(n-1);
6     int slast = f(n-2);
7     return last + slast;
8 }
9 int main() {
10 #ifndef ONLINE_JUDGE
11     freopen("input.txt", "r", stdin);
12     freopen("output.txt", "w", stdout);
13 #endif
14     cout << f(4);
15
16     return 0;
17 }
```

[Finished in 1.6s]

input.txt

```
1 5
2 1 2 3 4 5
```

output.txt

```
1 0
```

Line 17, Column 2; Build finished Tab Size: 4

TUF

The image shows a Sublime Text editor window with the following tabs:

- code.cpp
- stdc++.h
- input.txt
- output.txt

The code in `code.cpp` is as follows:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int f(int n) {
4     if(n <= 1) return n;
5     int last = f(n-1);
6     int slast = f(n-2);
7     return last + slast;
8 }
9 int main() {
10 #ifndef ONLINE_JUDGE
11     freopen("input.txt", "r", stdin);
12     freopen("output.txt", "w", stdout);
13 #endif
14     cout << f(4);
15
16     return 0;
17 }
```

The `input.txt` file contains the number 5.

The `output.txt` file contains the number 3.

The status bar at the bottom left says "[Finished in 1.6s]".

The status bar at the bottom right shows "Tab Size: 4".

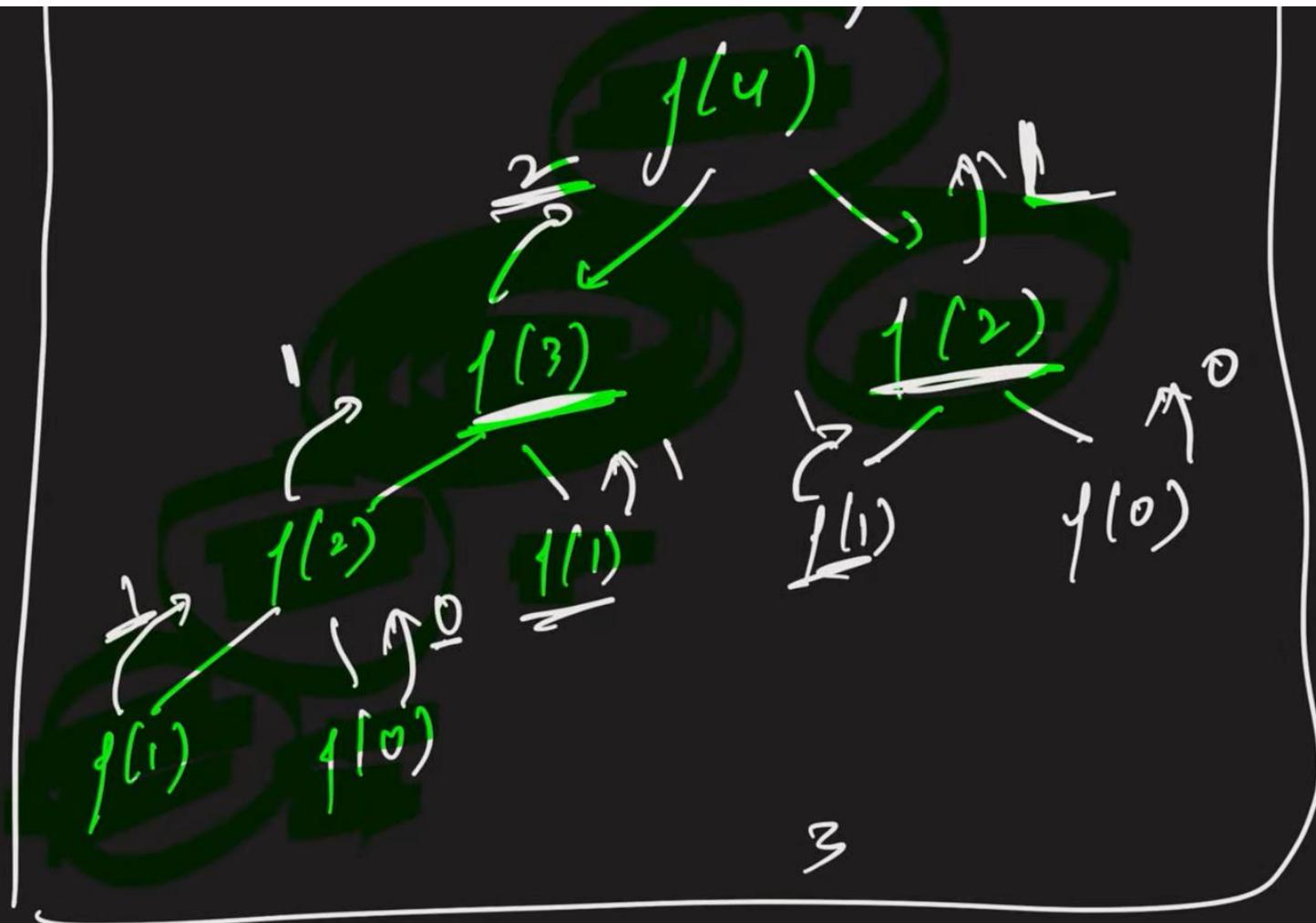
A video feed of a person with dark hair and a beard, wearing a blue shirt, is overlaid on the right side of the screen. The TUF logo is visible in the bottom right corner of the video feed area.

```
code.cpp      stdc++.h
1 #include<bits/stdc++.h>
2 using namespace std;
3 int f(int n) {
4     if(n <= 1) return n;
5     int last = f(n-1);
6     int slast = f(n-2);
7     return last + slast;
8 }
9 int main() {
10 #ifndef ONLINE_JUDGE
11     freopen("input.txt", "r", stdin);
12     freopen("output.txt", "w", stdout);
13 #endif
14     cout << f(4);
15
16     return 0;
17 }
```

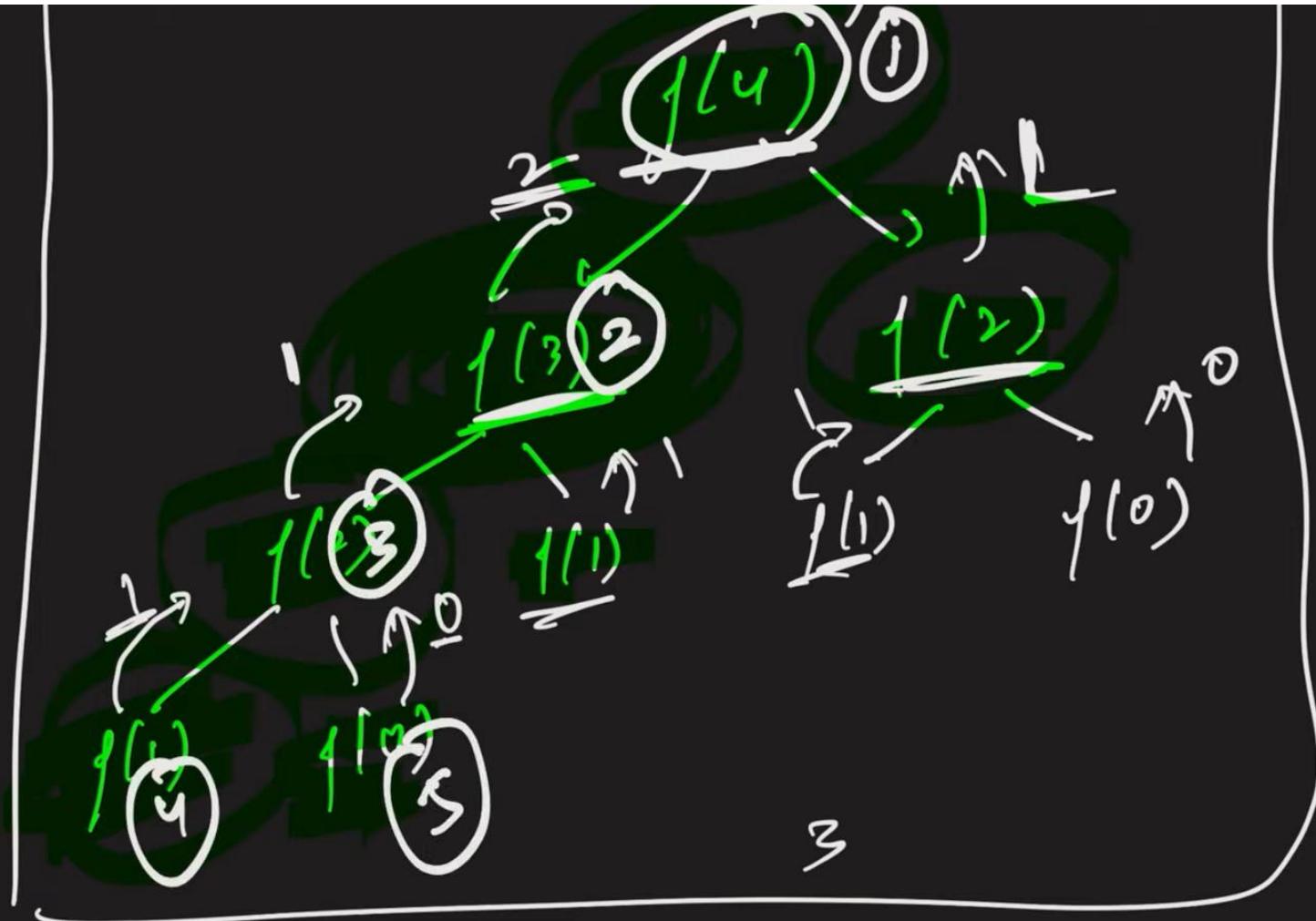
[Finished in 1.6s]

```
input.txt      x
1 5
2 1 2 3 4 5
output.txt    x
1 15
```

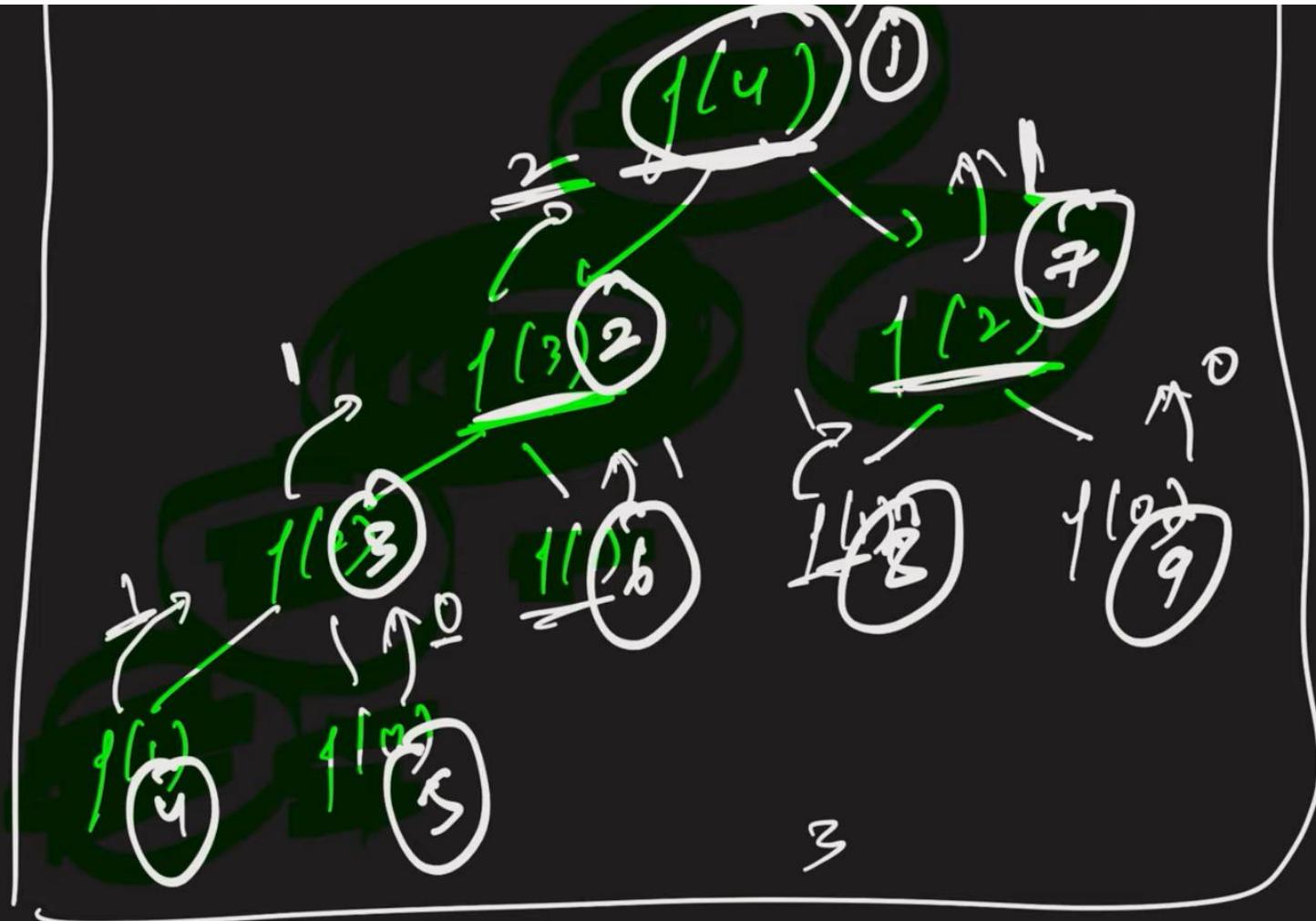
TUF



TUF

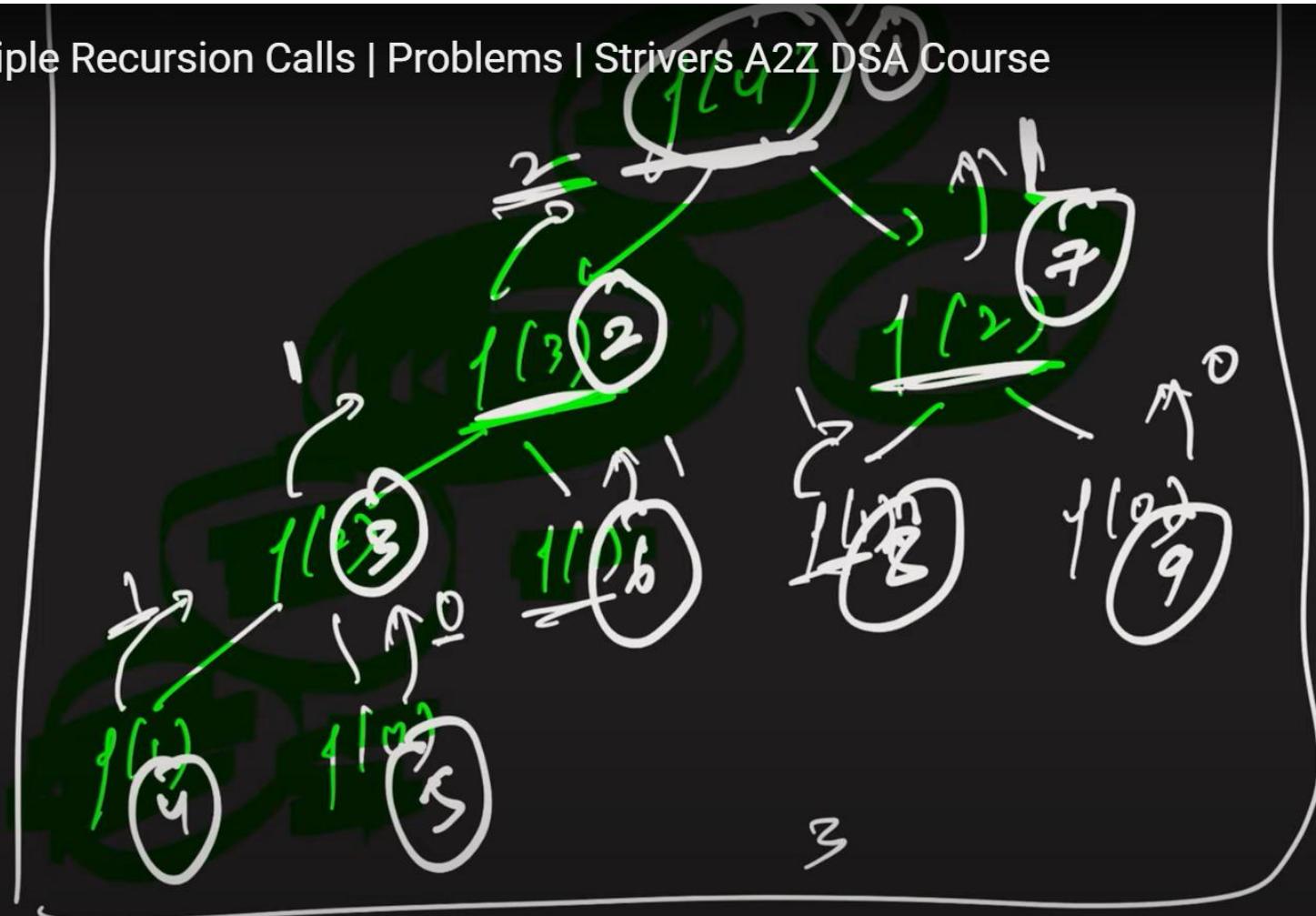


TUF



TUF

Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



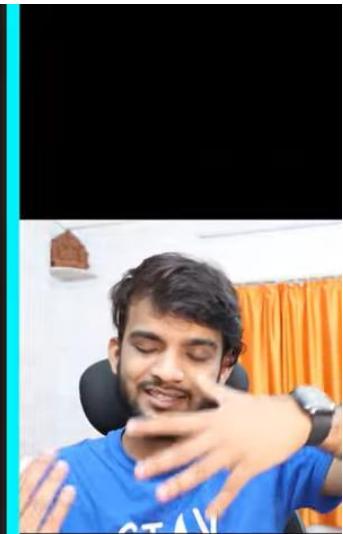
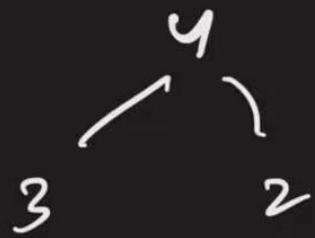
TUF

$\frac{1}{n}$



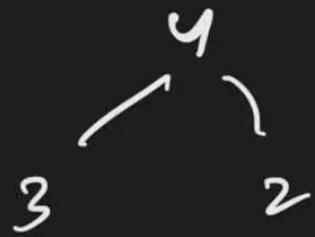
TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



$T \in \Theta(2^n)$

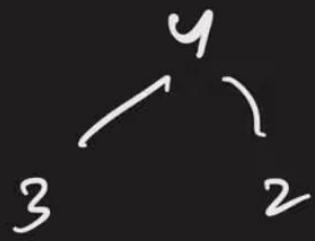


≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course

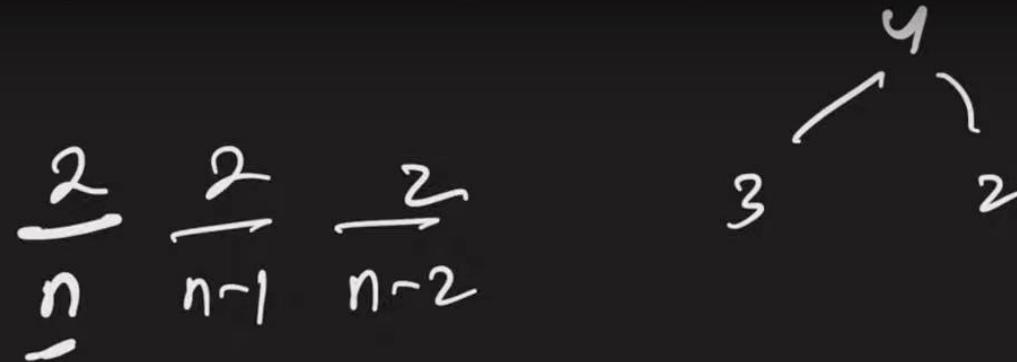


TUF

$$\frac{2}{n} \quad \frac{2}{n-1}$$

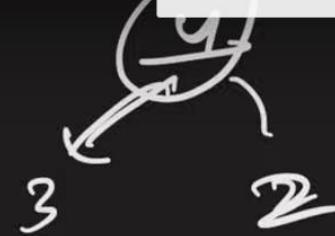


≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



TC $\rightarrow 2^n$



≡ Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



TC $\rightarrow (2^n)$



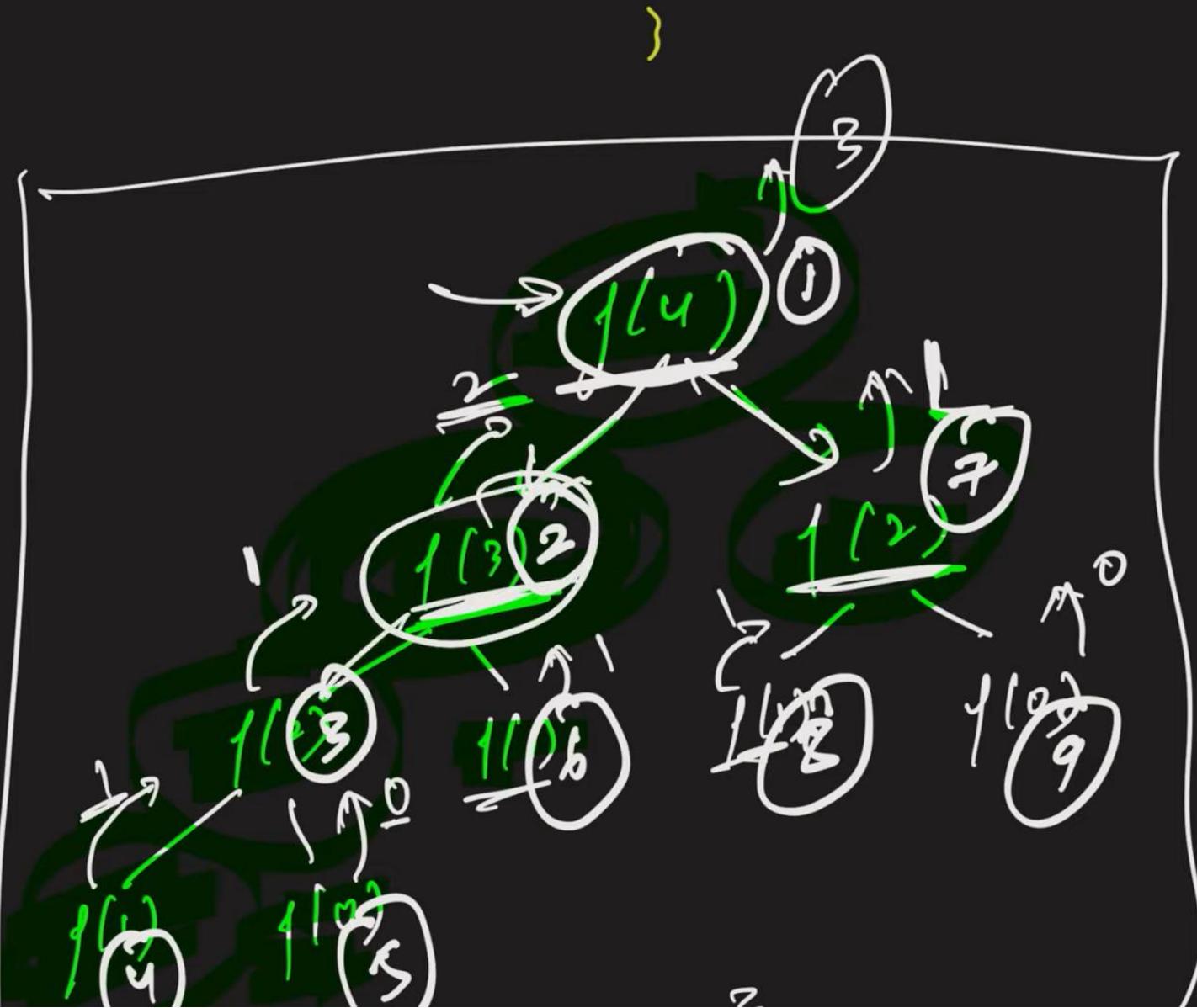
TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



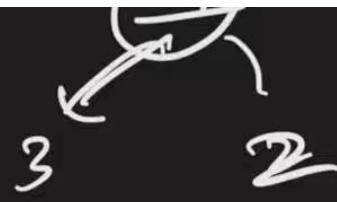
$T \propto (2^n)$ exponential





TUF

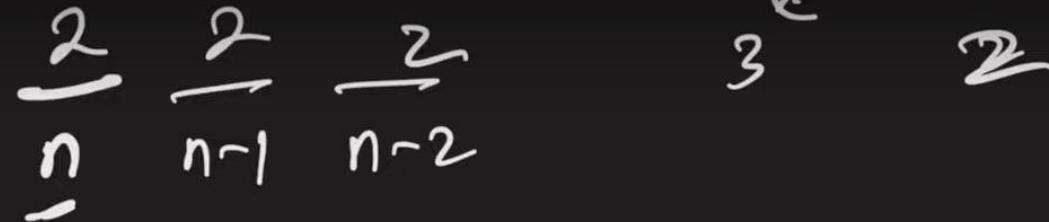
$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



$T \propto (2^n)$ exponential



Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



$T C \rightarrow (2^n)$ exponential

$y \rightarrow$

$$2^y = 16$$

TUF

Re 5. Multiple Recursion Calls | Problems | Strivers A2Z DSA Course



$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$

3 2

TC $\rightarrow (2^n)$ exponential

$y \rightarrow$

$$2^y = 16$$

$$\textcircled{9}$$



TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



$$TC \rightarrow (2^n)$$

exponential

$y \rightarrow$

$$2^4 = 16$$



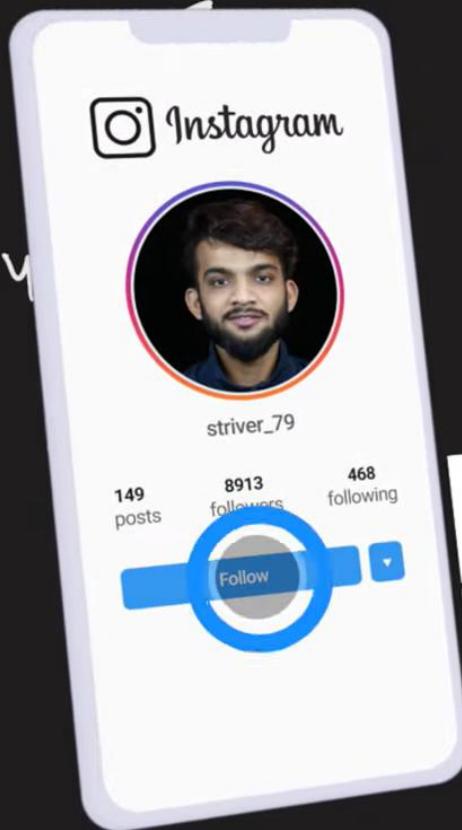
TUF

$$\frac{2}{n} \quad \frac{2}{n-1} \quad \frac{2}{n-2}$$



$$TC \rightarrow (2^n)$$

exponential



Y
@STRIVER_79
RAJ



Binary Trees | Binary Search Trees | C++
| Java | Data Structures and Algorithm...

54 videos

Striver's Graph Series | Playlist for
people who have limited time TO

GRAPH SERIES

30 videos



$$c \rightarrow (2^n)$$

exponential

TUF

