


Recursion → ? what is Recursion?

How to Study ?

↳ Peterson →



Interactive :- ()-

Function calls itself:-



Bookish
Language ~~X~~

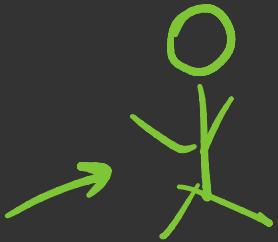
int solve ()



solve()



sundar (Life goals)



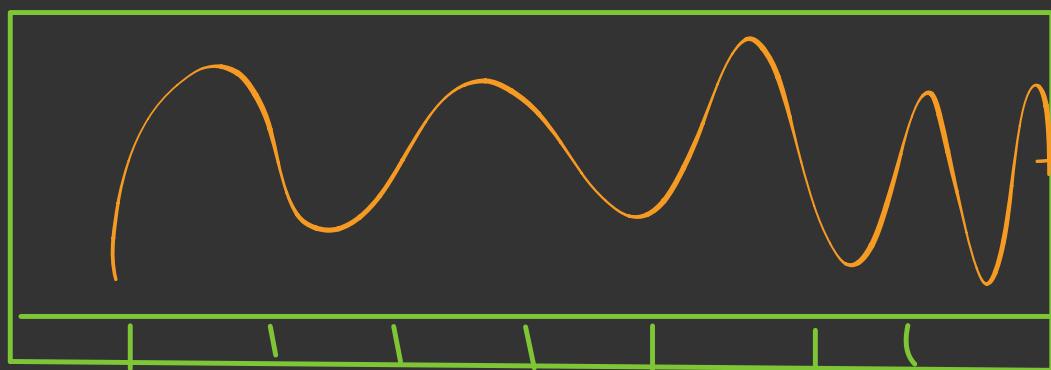
1 step

sikha dia

Sundari



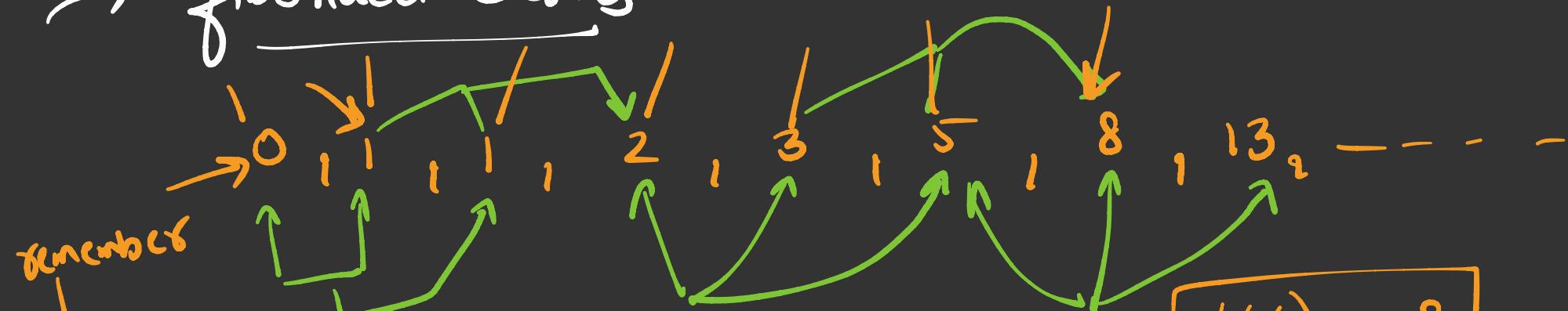
| case we need to solve
| but → recursion
| will take care



recursion

1 case

→ fibonacci Series :-



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

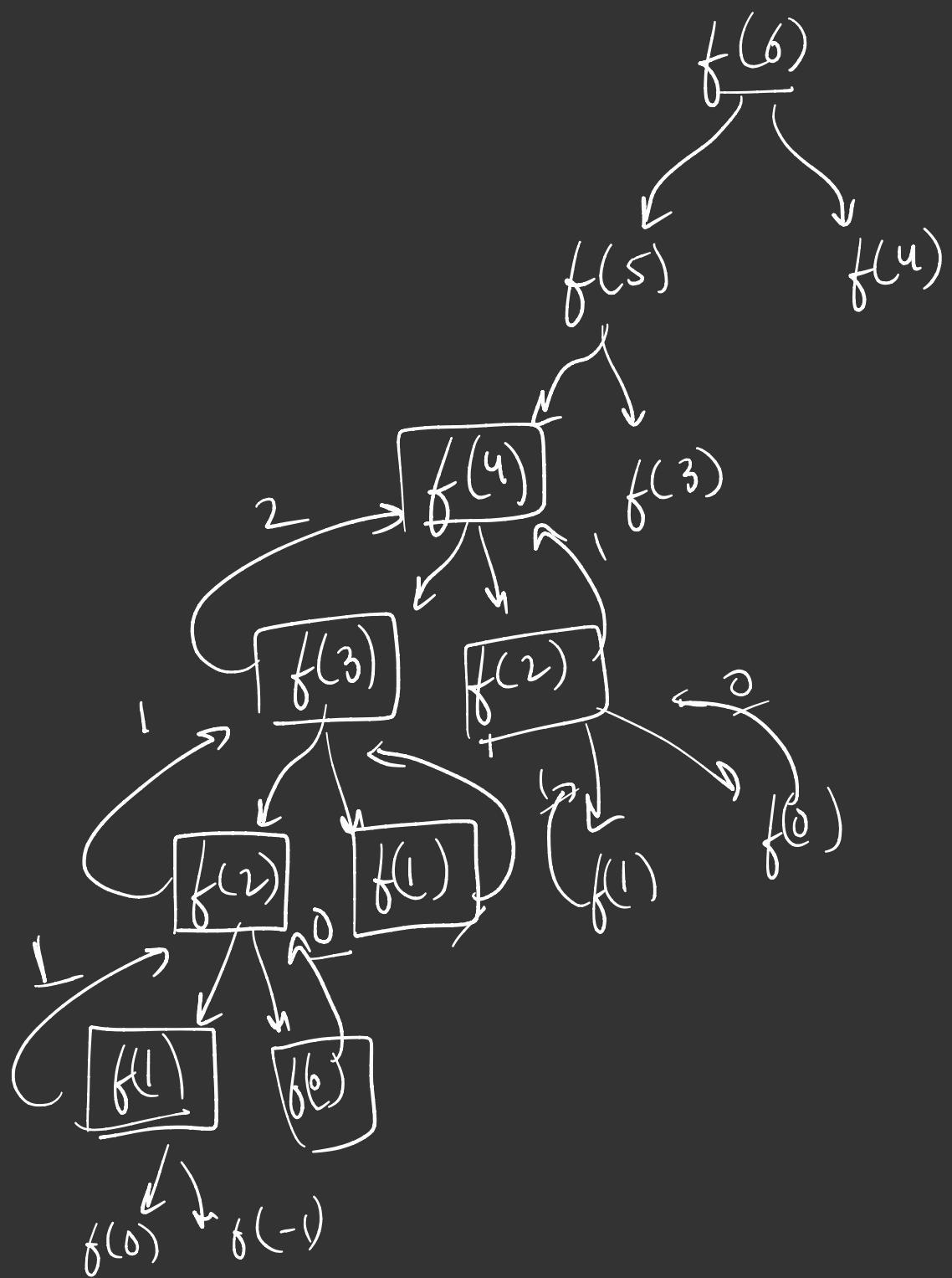
$\beta \omega$

$f(n) \rightarrow n^{\text{th}} \text{ term}$

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

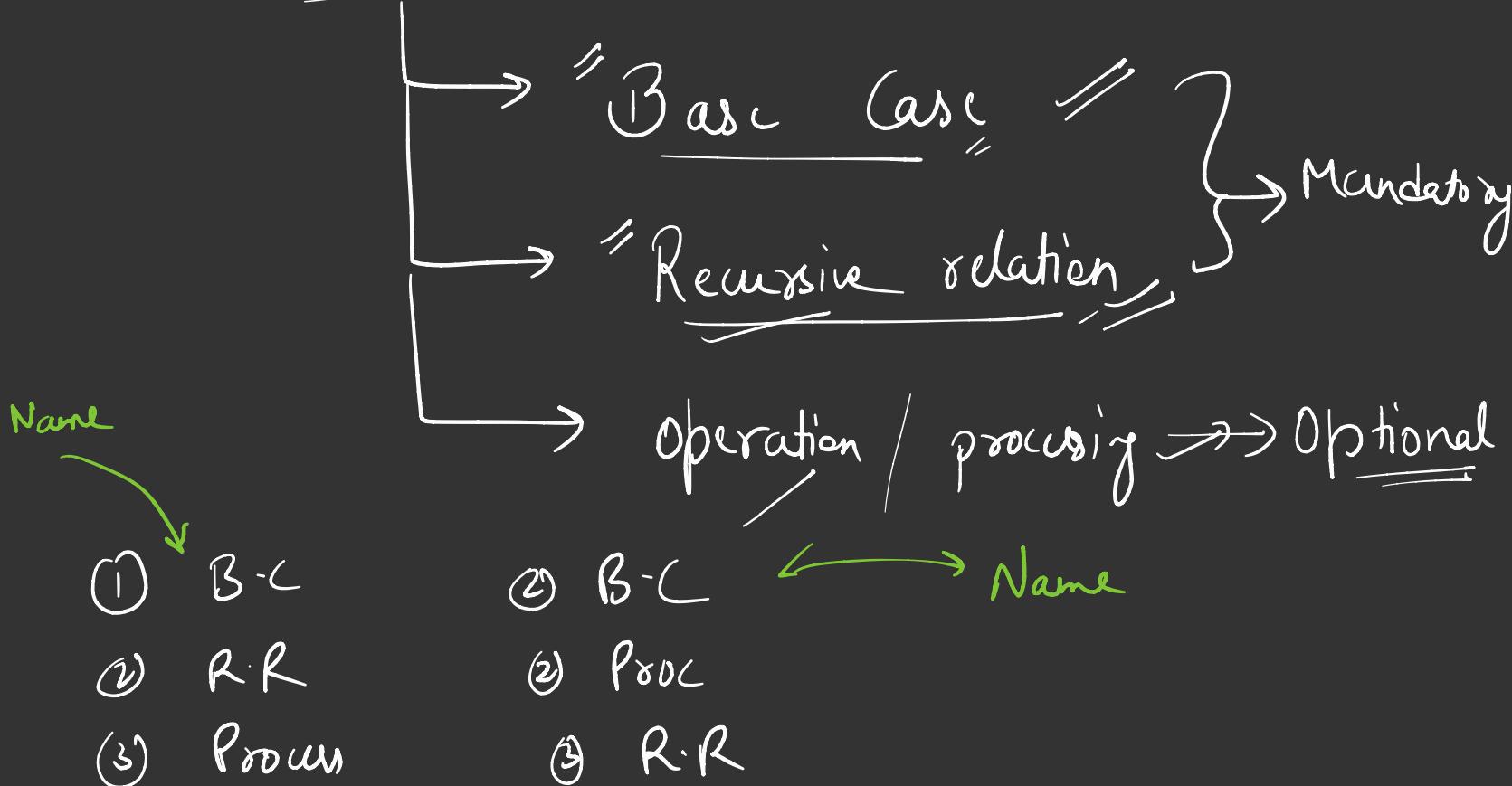
$$f(n) = \underline{f(n-1)} + f(n-2)$$

$$f(6) = f(5) + \underline{f(4)}$$





Recursion



Fib

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

Code:

```
int getFib (int n)
{
    // Base Case
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
```

$$\boxed{f(6) = f(5) + f(4)}$$

\downarrow \downarrow
 R_2 R_{rc}

int ans = getFib(n-1) + getFib(n-2);

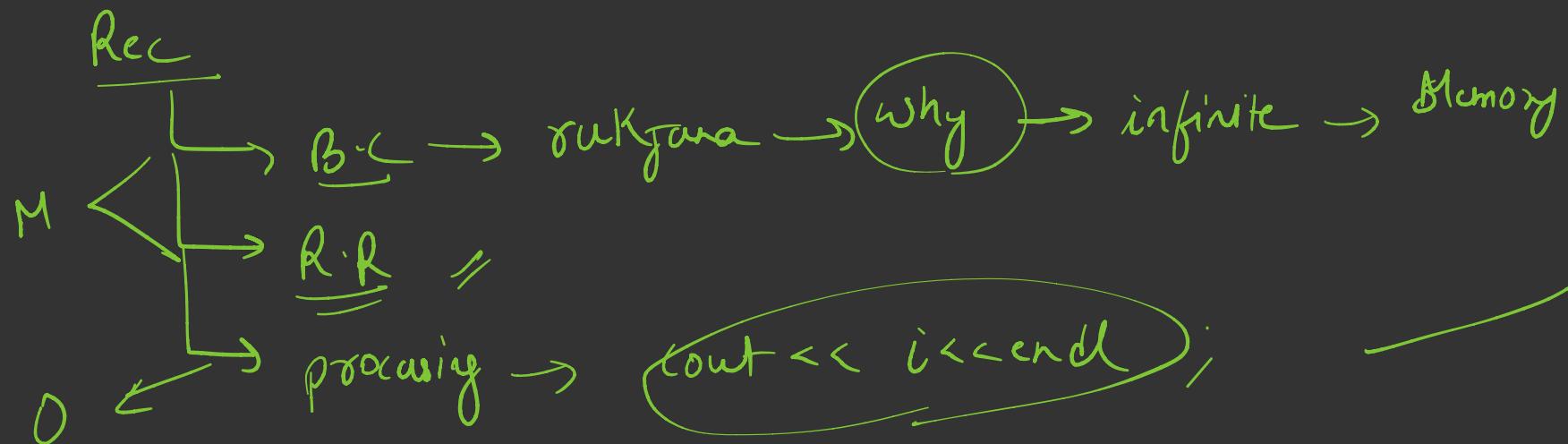
return ans;
}

Rec → function calls itself

(code)

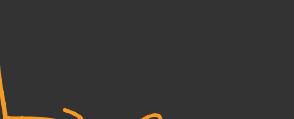
| calc solve → But → recursion

will take care



What is Recursion?

① what is Recursion? [Rec & BT]



Bookish



function calls itself

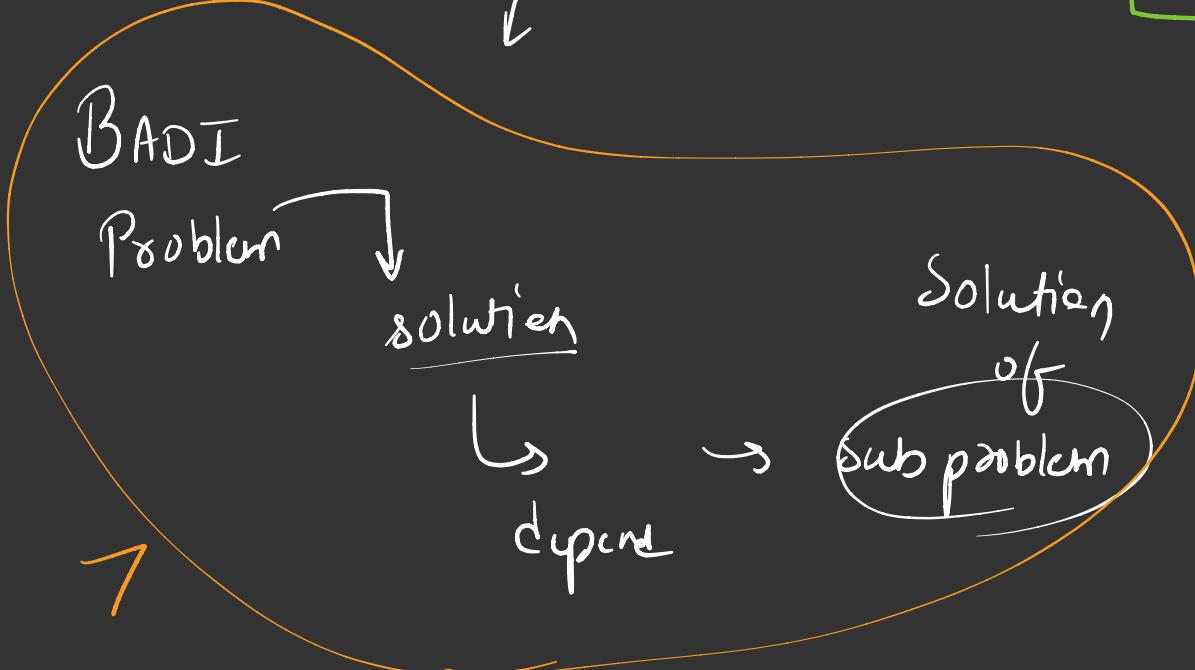


int solve (int n)

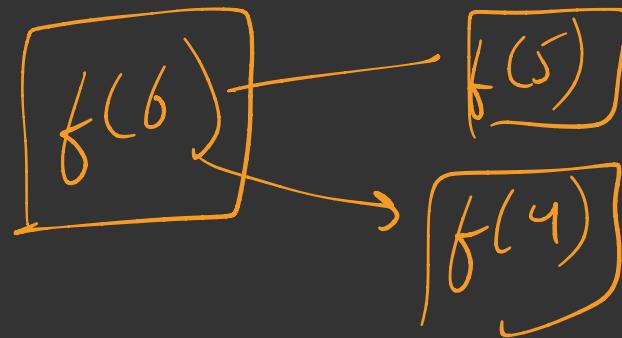


solve (n-1);

Recursion



apply recursion

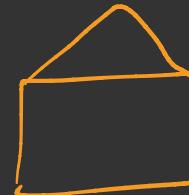
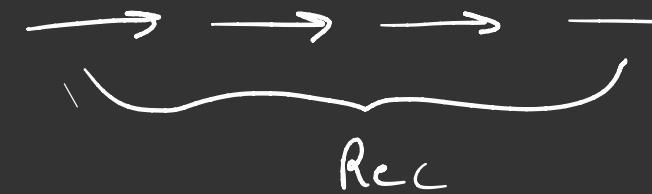


Kanha



src

humnc



dat

$n \rightarrow p$
print =
 $f(n-1)$

$f(n) \rightarrow$ point decreasing
order from n to 1

①

Point Decreasing

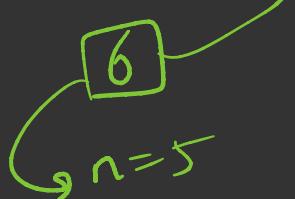
i/p $\rightarrow n = 6$

o/p \rightarrow



$cout << 6;$

$n = 6$



$n = 4$



$n = 3$



$n = 2$



$n = 1$



```

void solve ( int n )
{
    if (n == 0)
        return;
    cout << n << endl; → PFO
}

```

solve(n-1); → R.F

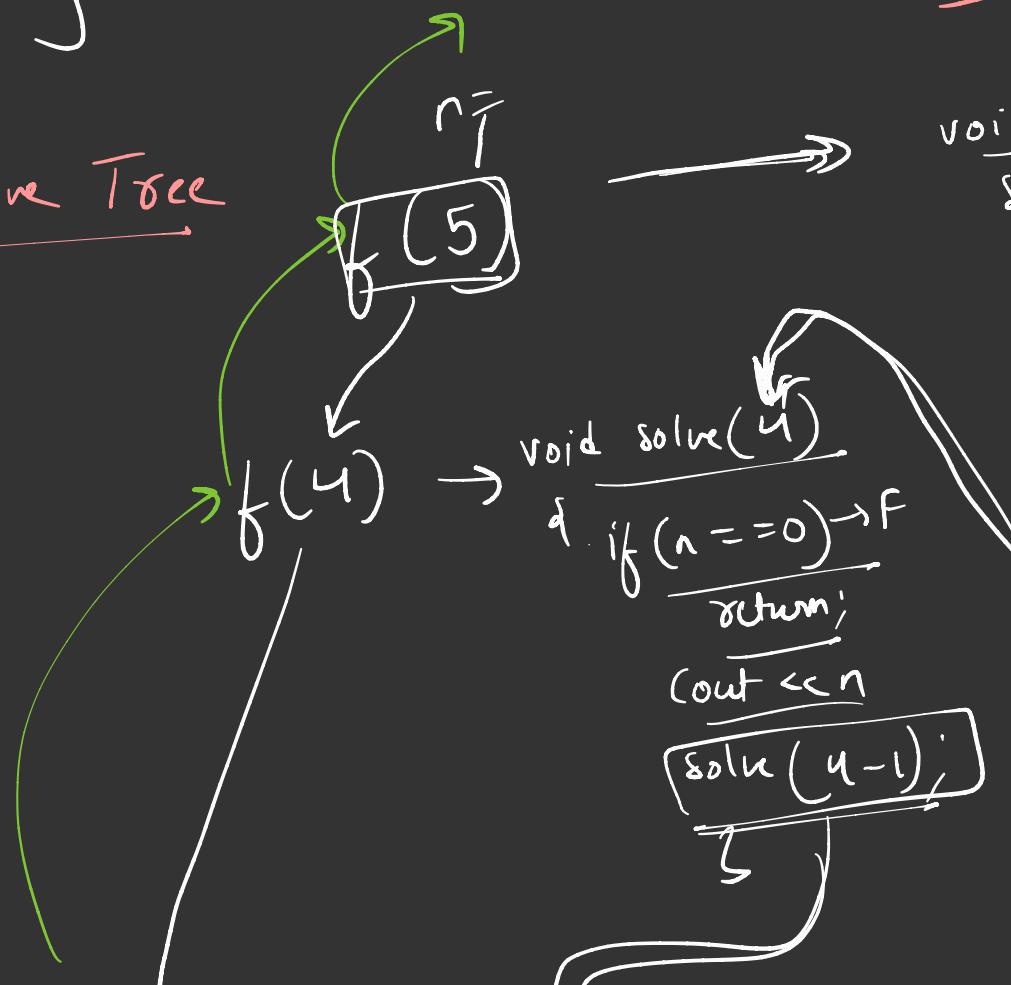
}

n=5

5 4 3 2 1

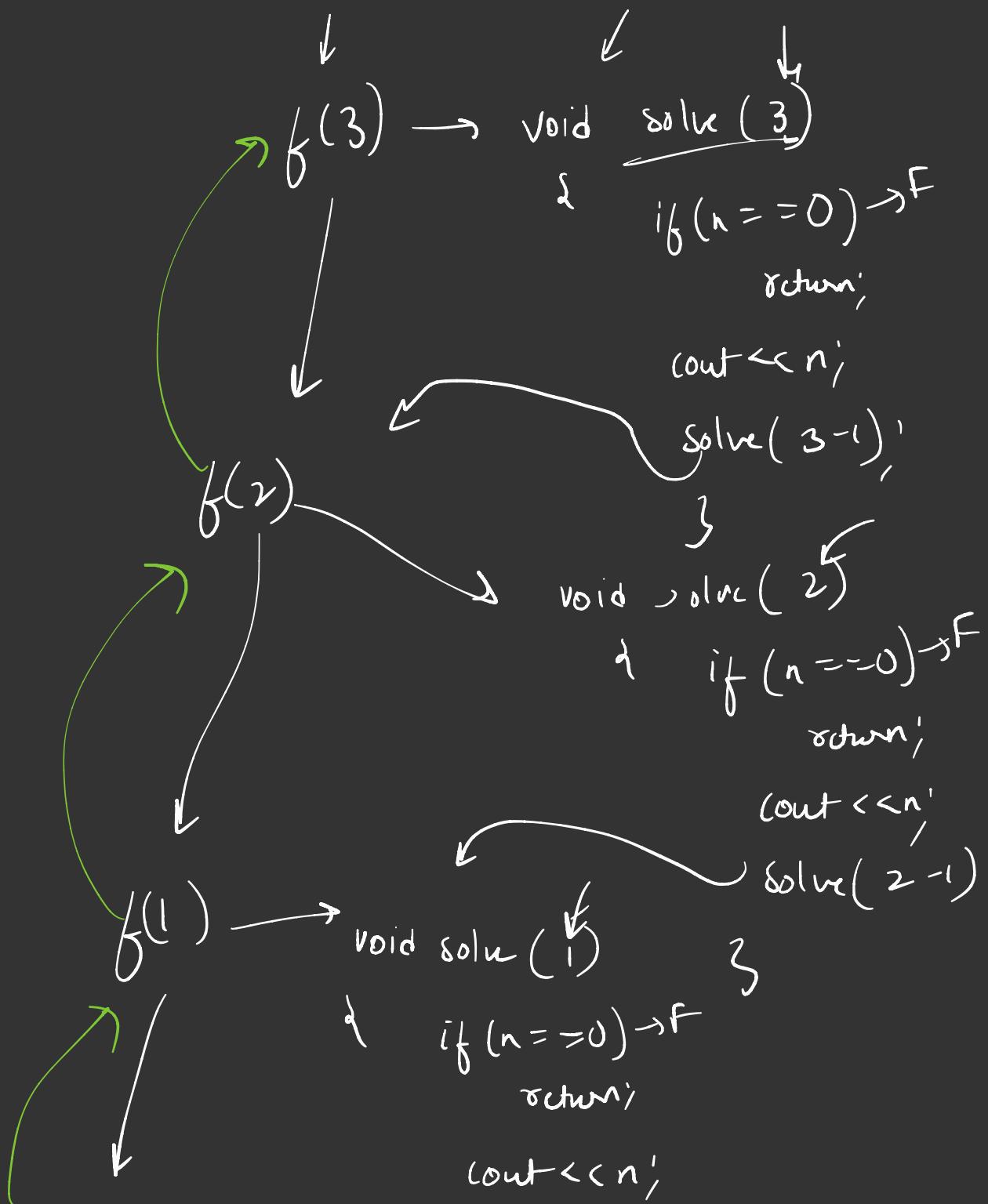
n=6 → Recursive Tree

if (n <= 0) → F
return;



void solve(5)
 if (n == 0) → F
 return;
 cout << n;
 solve(5-1);
}

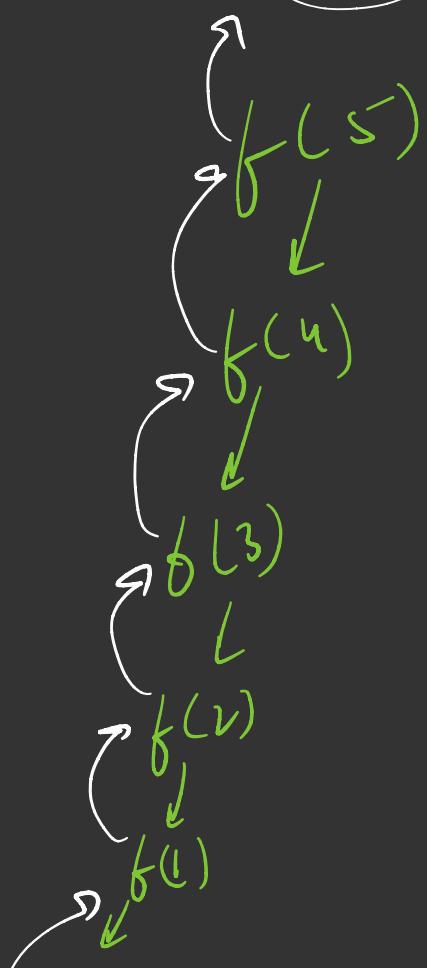
SO - 60 LPA





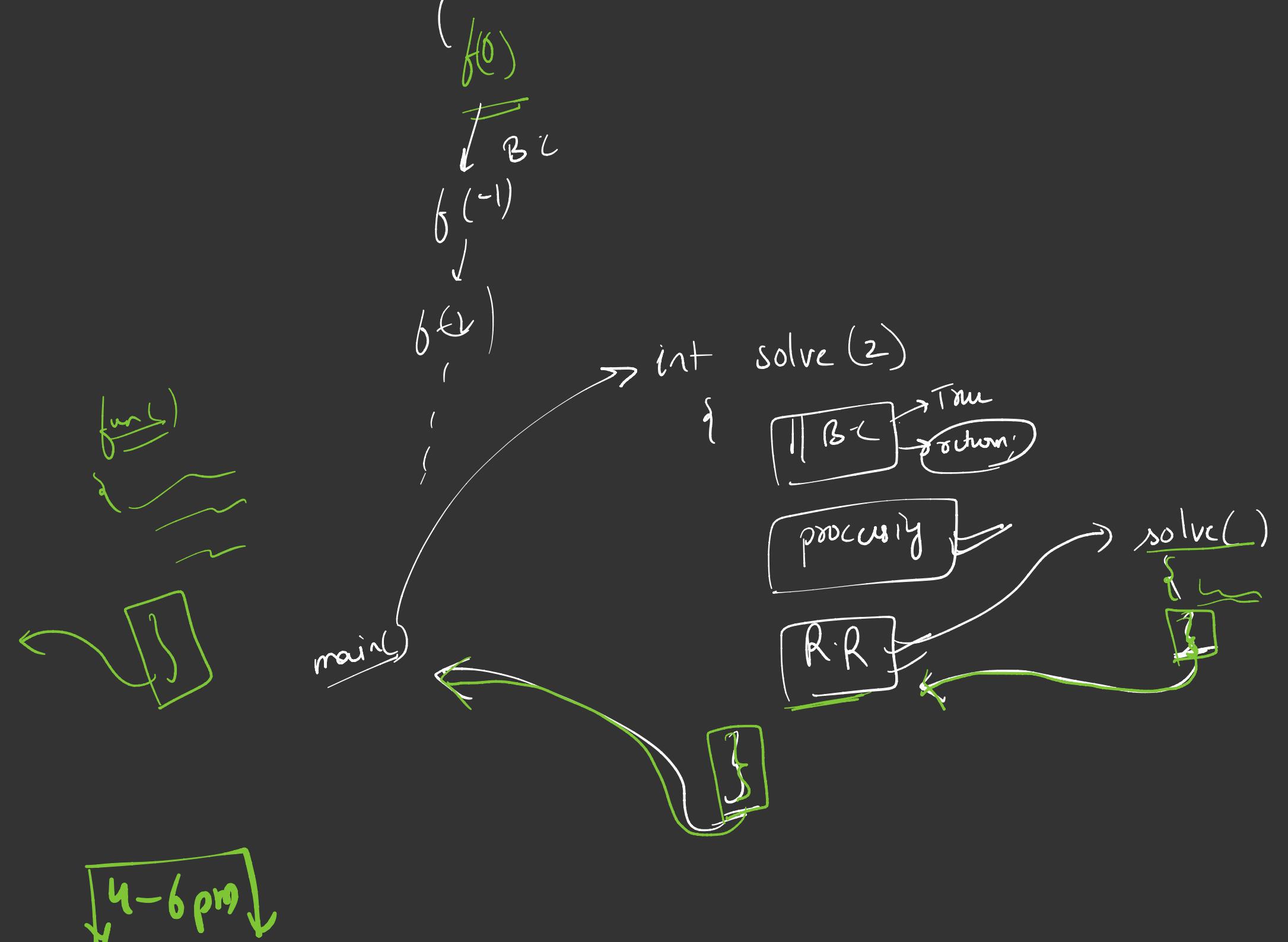
Rec tree

$f(0)$ ↘
 \downarrow $solve(1-1)$
 \downarrow
 \downarrow void $solve(0)$
 $\quad \& \quad$ if ($n == 0$) → True
 $\quad \quad \quad$ return;



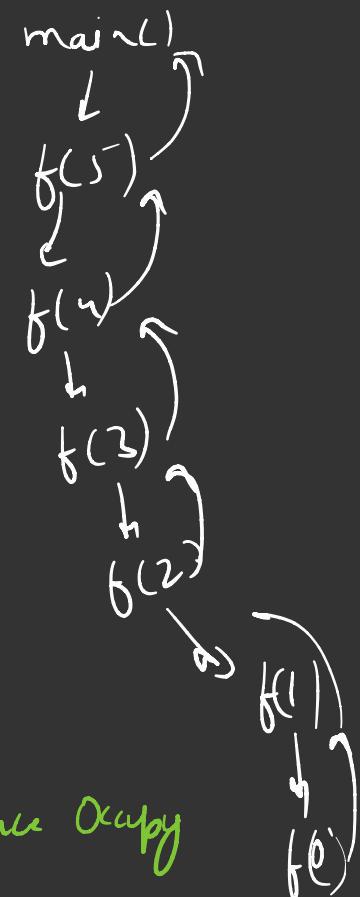
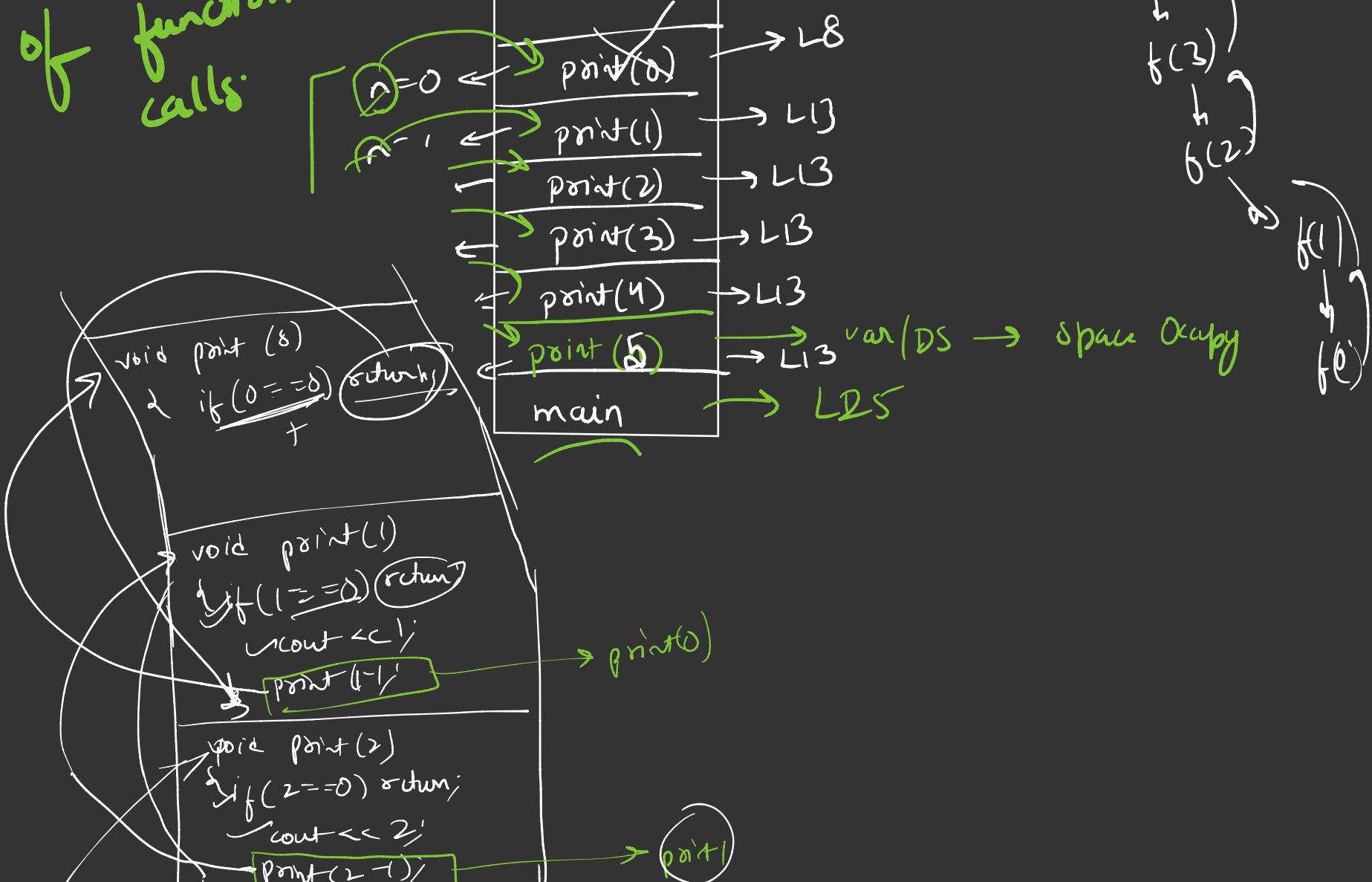
TC | SC

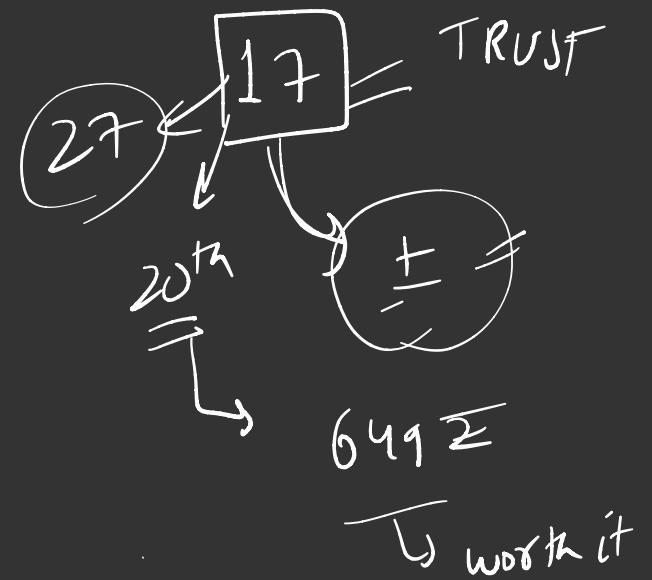
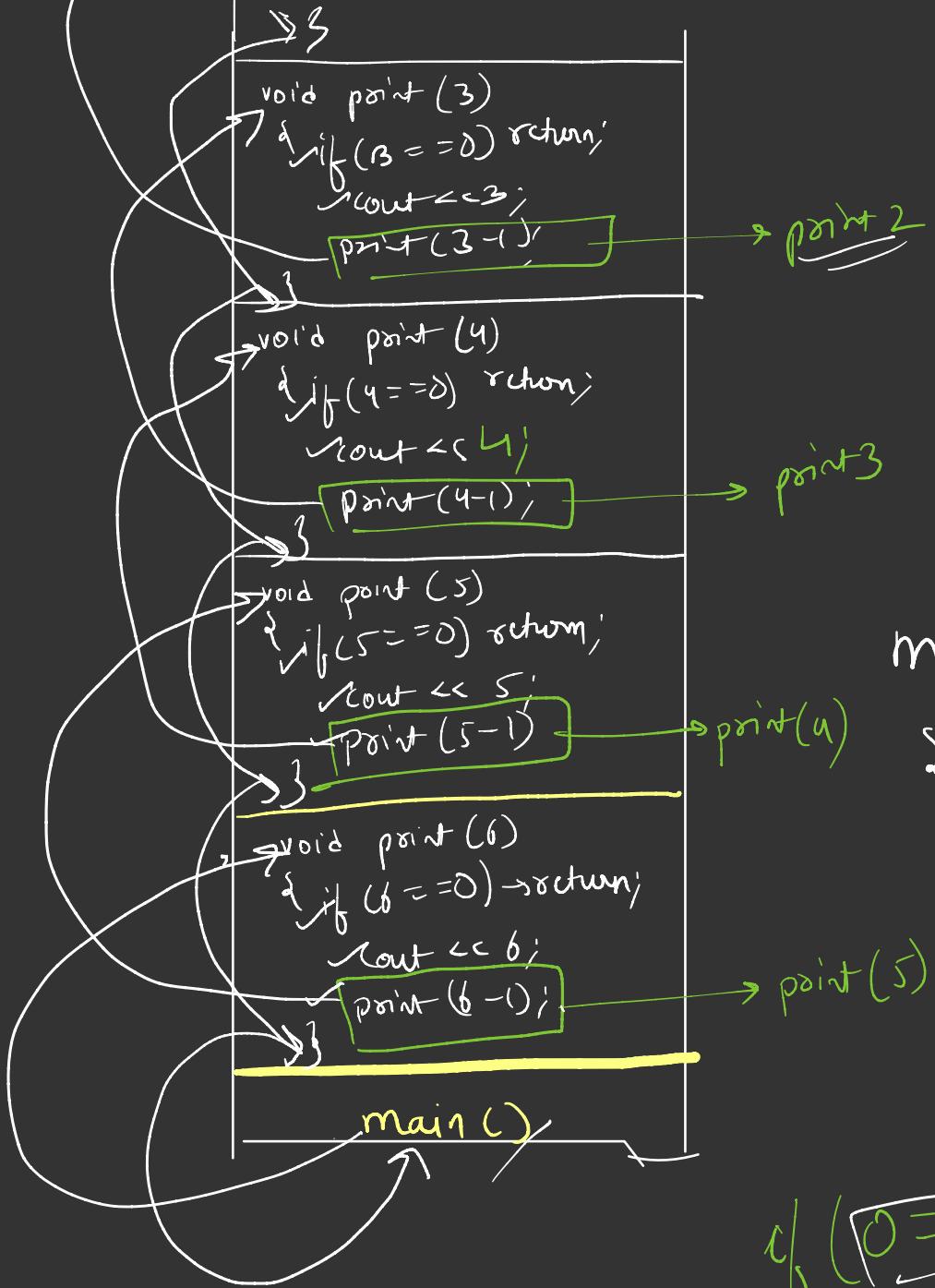
- ① what is Recursion?
- ② Rec tree
- ③ call stack



Call stack → ?

Keeps track
of function
calls.





main()

→ point(6)

point(5)

point(4)

if (`i == 0`)
return;

what is Rec?

Rec tree

Call Stack



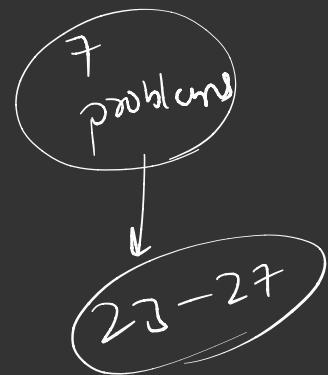
TRUST

6492

↳ work it

①

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

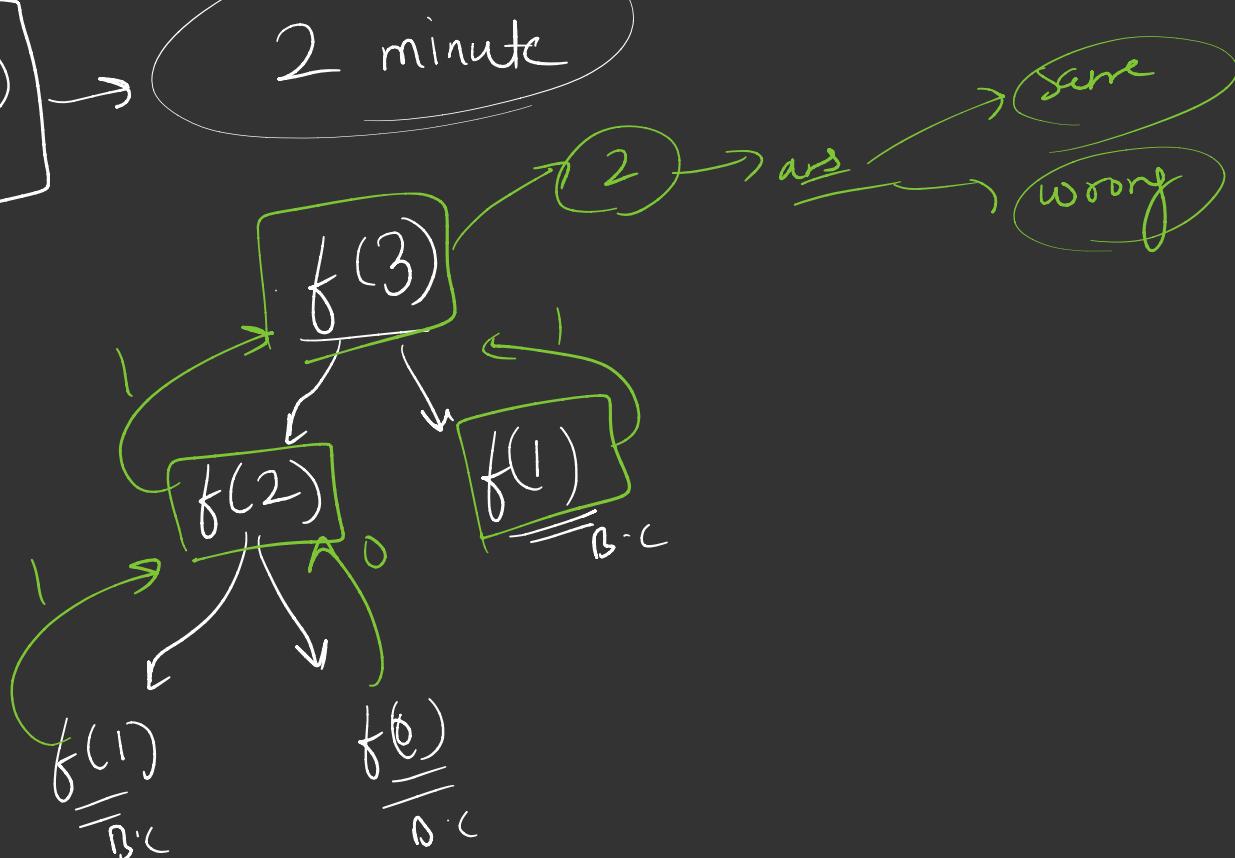


B.C. \rightarrow

$$\begin{cases} f(0) = 0 \\ f(1) = 1 \end{cases}$$

Rec
tool $\rightarrow f(3)$

2 minute

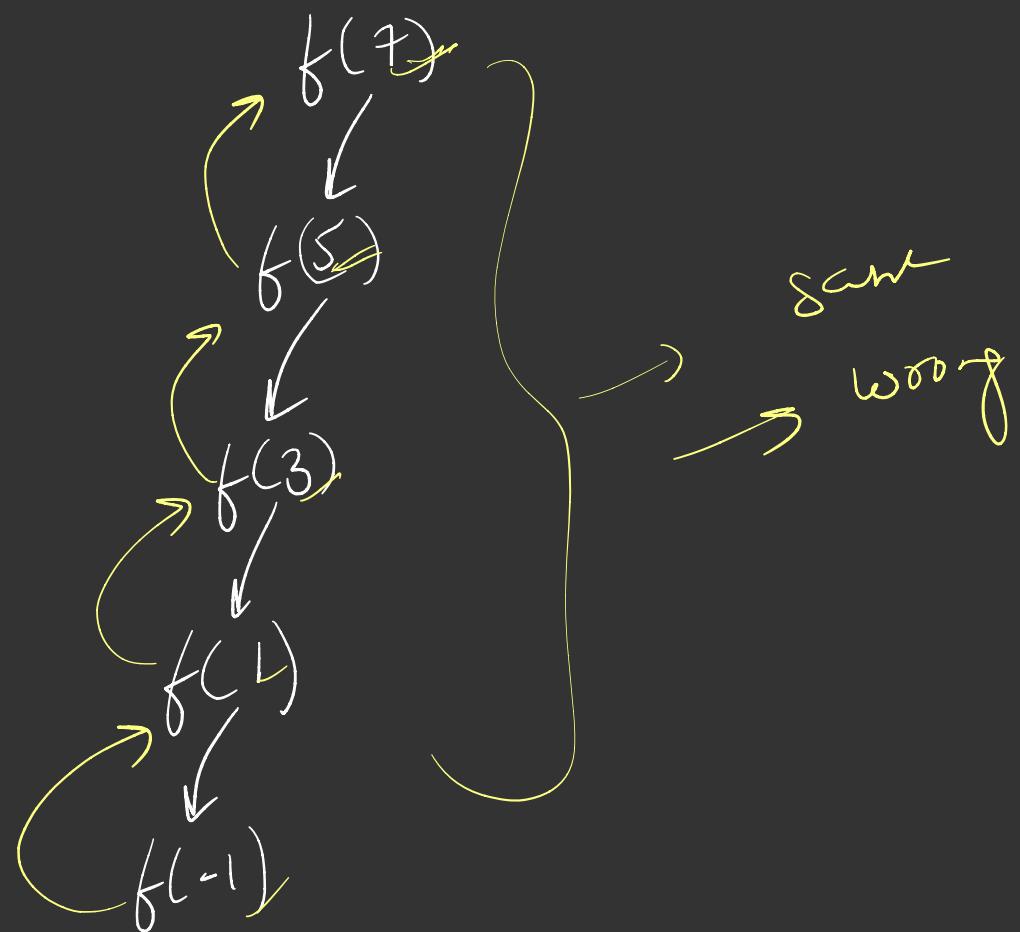


$$f(n) \Rightarrow f(n-2)$$

$$n < 0 \Rightarrow a_0 = 0$$

$$n = 0 \Rightarrow a_0 = 1$$

$f(7)$ → Rec tree? → bangga
 → barakah

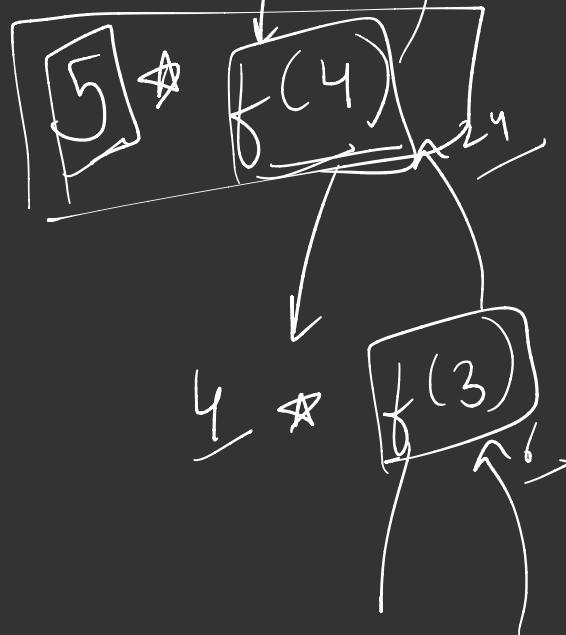
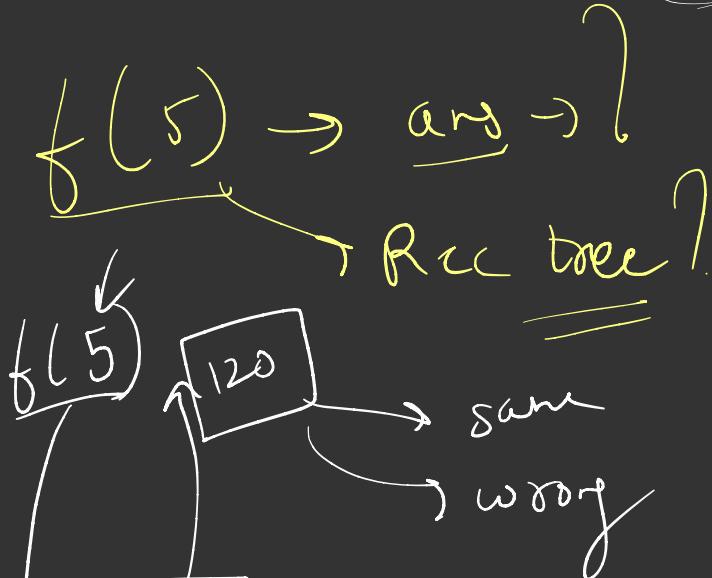


$$f(n) = n * f(n-1)$$

↑ factorial

$$f(n) = n * f(n-1)$$

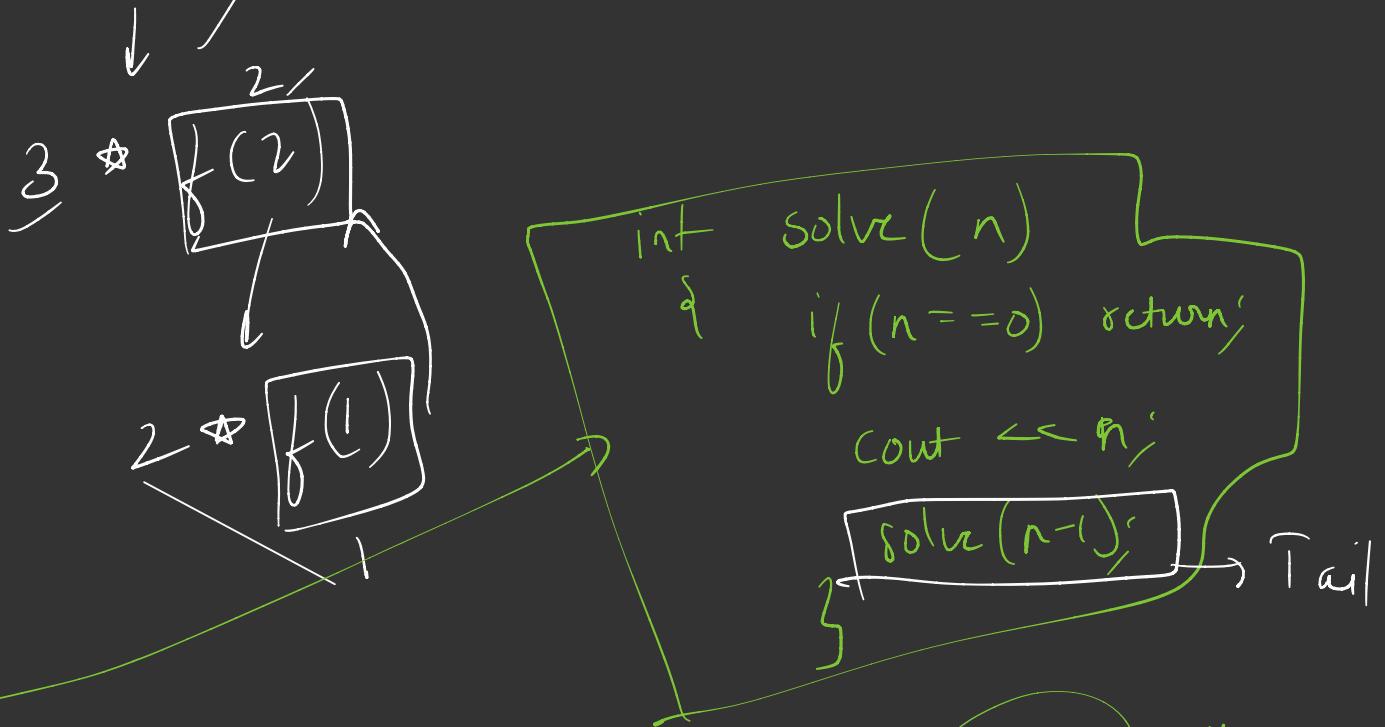
$$\begin{aligned} f(5) &= 5 * f(4) \\ &= 5 * 4 * f(3) \end{aligned}$$



B.C

```
if (n == 1)
    return 1;
```

32 sec



① Point Dec //

② Point Inc: //

i/p \rightarrow $n = 5$

o/p \rightarrow $1, 2, 3, 4, 5$

1 min \rightarrow How to Solve?

Know

```

void
{
    solve (int n)
    {
        if (n==0)
            return;
        solve (n-1);
        cout << n;
    }
}

```

1 2 3 → 0 | p

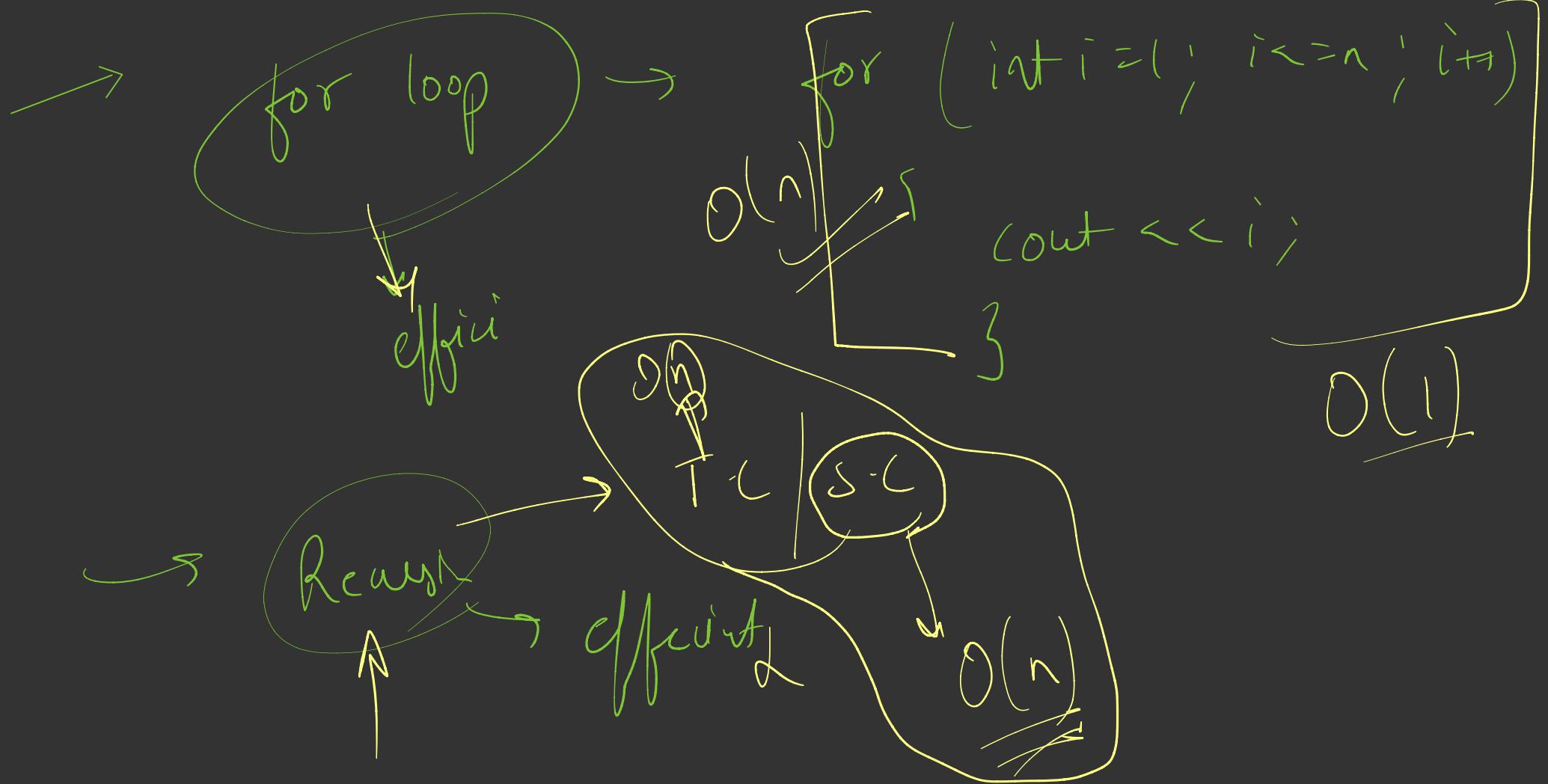
```

solve(3)
void solve(3)
{
    if (3 == 0) → F
    return;
    solve(3-1)
    cout << n;
}

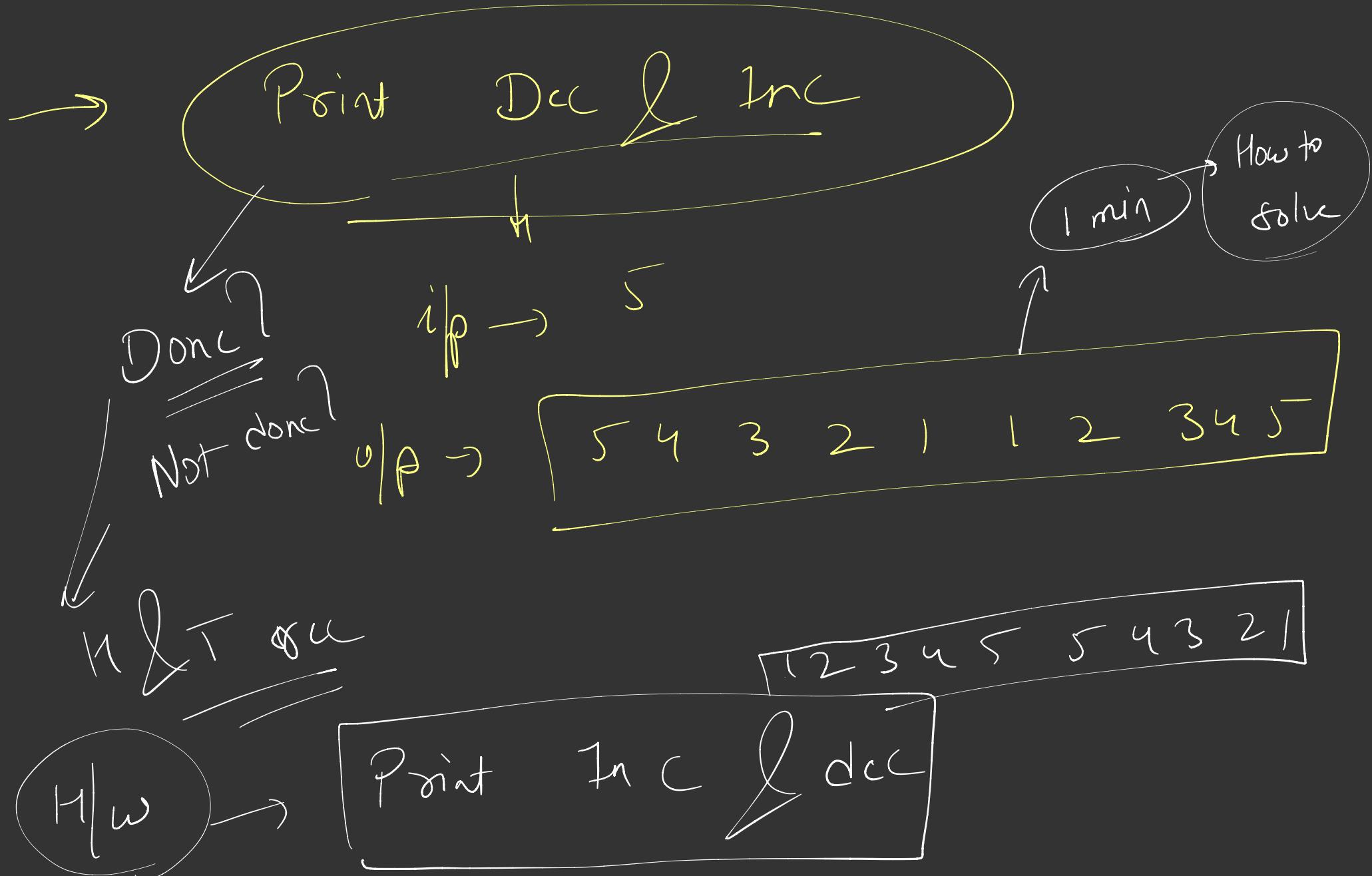
3
solve(2)
if (2 == 0) → F
return;
solve(2-1)
cout << 2;
3

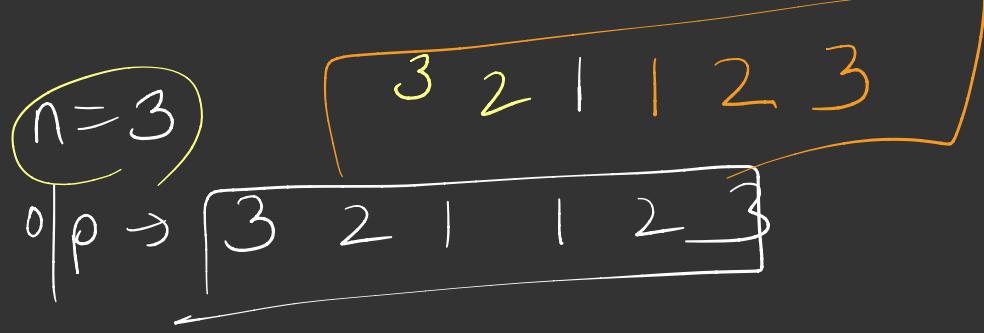
void solve(1)
if (1 == 0) → F
return;
solve(1-1)
cout << 1;
3

```



→ Point Dec / → Point Inc /





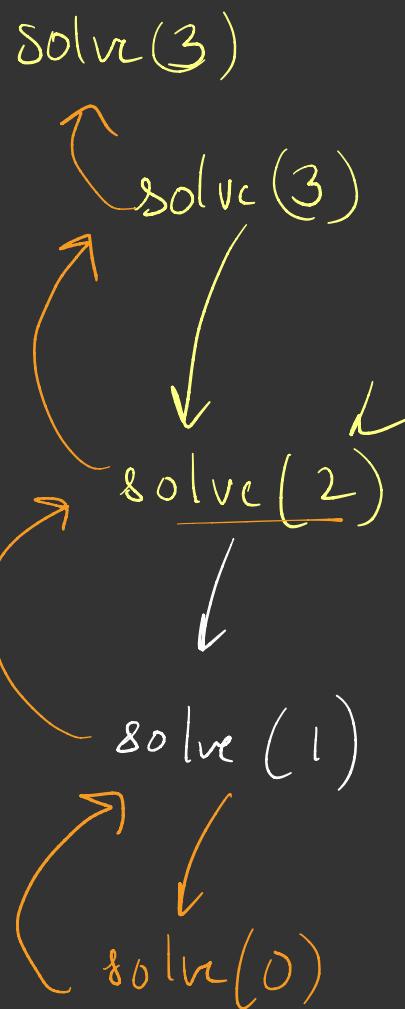
~~void~~
~~solve~~
solve (int n)

}
 if ($n == 0$)
 return;

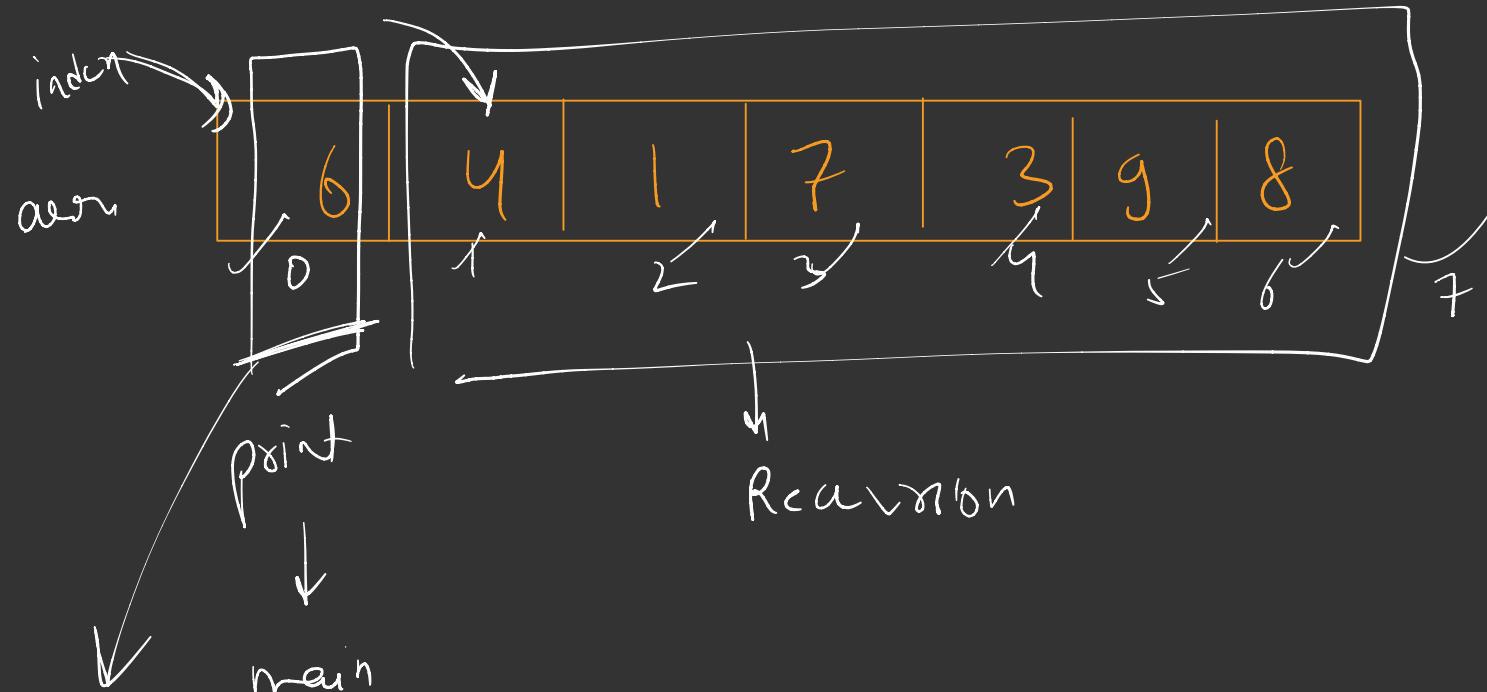
cout << n;

solve(n-1);

cout << n;

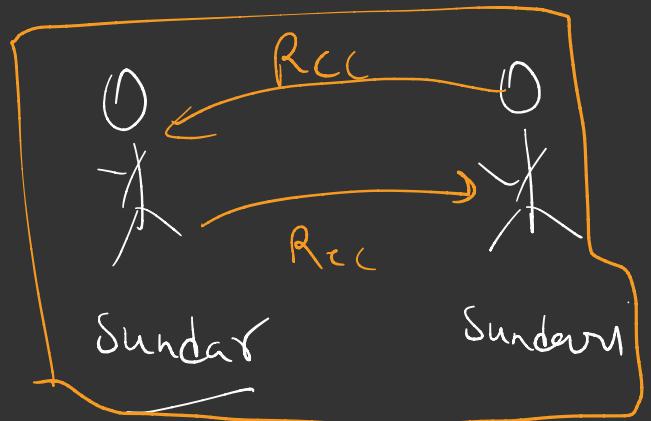
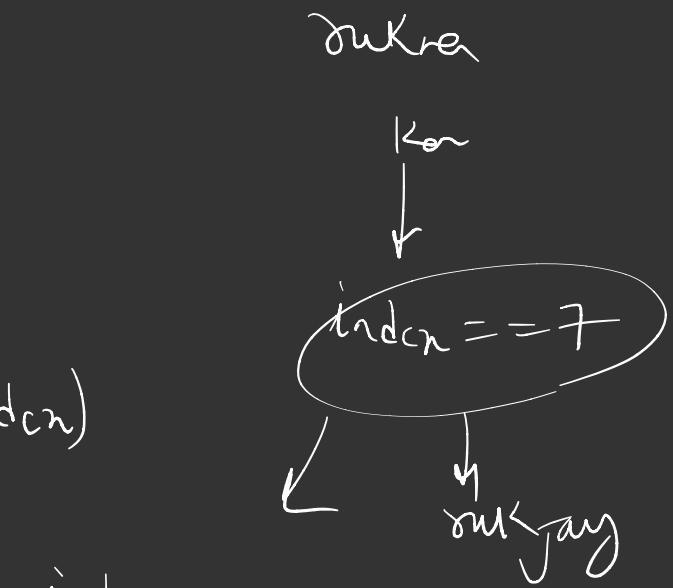


indent |



```

Void print ( int arr[], int size , int index)
{
    || Q-L
    if ( index == size )
        return;
    cout << arr [index];
    print ( arr , size , index + 1 );
}
    
```



→ factorial:-

$$5! = \underset{\text{min}}{5 \times} \underset{\text{Rec}}{[4 \times 3 \times 2 \times 1]} \quad \text{B.C}$$

$$= 5 \times 4!$$

$\text{P} = 0$ → stop return

$f(n) \rightarrow$ factorial of n

$$5! = 5 \times 4!$$

$$f(n) = n \star f(n-1)$$

Rec Relation

```
int factorial ( int n )  
{  
    // B.C  
    if (n == 0) X  
        return 1;  
    if (n == 1)  
        return 1;  
}
```

```
int ans = n * factorial(n - 1);
```

```
return ans;
```

```
}
```

BigInteger

$n >= 0$

$1! = 1$

$0! = 1$

Editor →

11/10

wrong answer

11-20

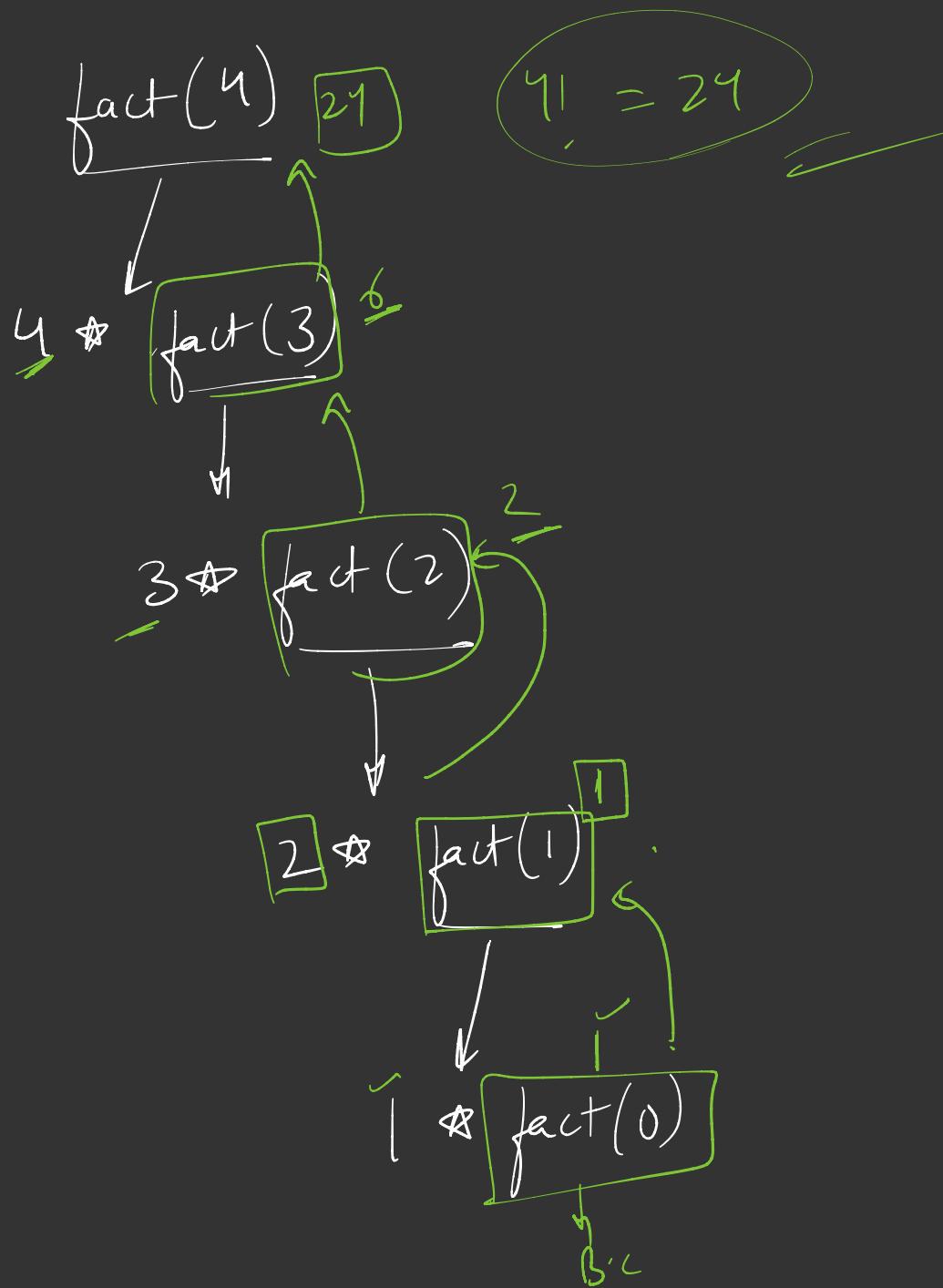
why

How to
resolve this

factorial(4)

C B HK

hadd sc aasaan
question



① what is Recursion?

glitch → 3-5 → errors

② Rec tree

③ call stack

④ Rec → visualize

⑤ Print Dec → 5 DRY RUN

⑥ ~ Inc

⑩ Array Point

⑦ ~ Dual Inc

⑧ Fibonacci

⑨ factorial