Q => Given a number n, print the first n natural numbers in increasing order recursively.

Ex → $n = 6$

Output → 1
2
3
4
5
6

$f(n) =$ ① $f(n-1)$ ② print (n)

can print first n, natural numbers recursively

if $(n<1)$ return;

Base Case

- if you've $n<1$
- we don't need to proceed

Assumption → let's assume function f works correctly for $n-1$. i.e. $f(n-1)$ correctly prints the first $n-1$ natural numbers for us.

Selfwork → print n.

Q ⇒ Given a +ve integer value (> 0) n. Count the no. of Binary Strings (Strings which only got 0 or 1) of length n, such that there are no consecutive ones.

Ex → $n = 3$

↳ ans → 5 → (000, 001, 010, 100, 101)

$n = 1 \rightarrow 2$

$n = 2 \rightarrow 3$

$n = 3 \rightarrow 5$

$n = 4 \rightarrow 8$

$n = 5 \rightarrow 13$

fibonacci

(0, 1)

(00, 01, 10)

(000, 001, 010, 100, 101)

(0000, 0001, 0010, 0100, 1000, 1001, 1010, 0101,

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

base case

if (n == 1) relurn 2;

if (n == 2) return 3;

chage

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \le i \le N$), the height of Stone i is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to Stone $i+1$ or Stone $i+2$. Here, a cost of $|h_i - h_j|$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Ex → [10, 30, 40, 20]

ans → 30

frog

10	30	40	20
1	2	3	4

→ 4 stones

Ex → [10, 10]

ans → 0

cost → |10 − 30| → 20

cost → |30 − 20| → 10

→ 30

Ex → [30, 10, 60, 10, 60, 50]

ans → 40

$\boxed{f(i, n)}$ ←

min cost

Frog

h_1	h_2	h_3	h_4	h_5	...	...	h_N
1	2	3	4	5	...	...	N

N

$\nearrow$ 0.H
1

we will try to explore all possibilities

from 1^{st} stone, frog can jump either to the 2^{nd} or 3^{rd} stone.

$1^{st} \to 2^{nd} \to |h_1 - h_2|$

$1^{st} \to 3^{rd} \to |h_1 - h_3|$

if we somehow get the minimum cost to reach N^{th} stone from 2^{nd} stone (x) & the min cost to reach N^{th} stone from 3^{rd} stone (y).

$$1^{st} \xrightarrow{|h_1 - h_2|} 2^{nd} \xrightarrow{x} N$$

or

$$1^{st} \xrightarrow{|h_1 - h_3|} 3^{rd} \xrightarrow{y} N$$

$$\min\left(|h_1 - h_2| + x,\ |h_1 - h_3| + y\right) \rightarrow \underline{\text{ans}}$$

$$f(i, n) = \min\left(|h_i - h_{i+1}| + f(i+1, n),\ |h_i - h_{i+2}| + f(i+2, n)\right)$$

$f(i,n)$ returns the min cost to reach n^{th} stone from i^{th} stone

$f(i+1,n)$: min cost to reach n^{th} stone from $(i+1)$

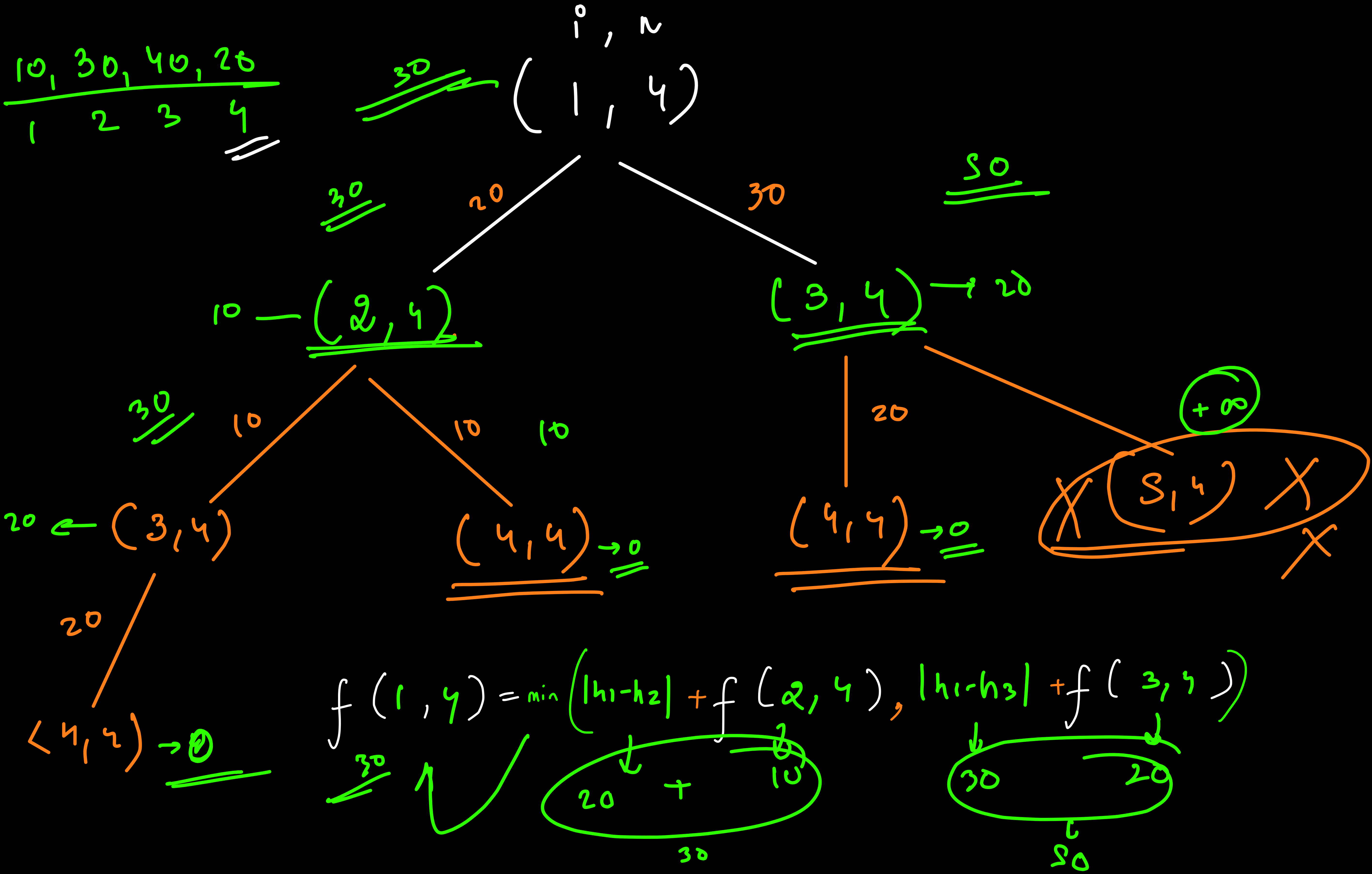
final ans $\rightarrow \underline{f(1, n)}$

$$i^{th} \rightarrow (i+1)^{th} \rightarrow |h_i - h_{i+1}|$$

$$i^{th} \rightarrow (i+2)^{th} \rightarrow |h_i - h_{i+2}|$$

assumption → assume that function f works correctly for i+1 and i+2 i.e. funcⁿ f correctly gives you min cost to reach N^{th} stone from $(i+1)^{th}$ stone & $(i+2)^{th}$ stone.

Selfwork → from i^{th} stone consider all possibilities and calculate min of it



$$f(1,4) = \min\left(|h_1 - h_2| + f(2,4),\ |h_1 - h_3| + f(3,4)\right)$$

20 + 10 = 30

30 + 20 = 50

```
f(i, n) {
    if (i == n) return 0;
    if (i > n) return ∞;
    minCostViaIplus1 = |h[i] - h[i+1]| + f(i+1, n);
    minCostViaIplus2 = |h[i] - h[i+2]| + f(i+2, n);
    return min(minCostViaIplus1, minCostViaIplus2);
}
```

There are N stones, numbered $1, 2, \ldots, N$. For each i ($1 \leq i \leq N$), the height of Stone i is h_i.

There is a frog who is initially on Stone 1. He will repeat the following action some number of times to reach Stone N:

- If the frog is currently on Stone i, jump to one of the following: Stone $i+1, i+2, \ldots, i+K$. Here, a cost of $|h_i - h_j|$ is incurred, where j is the stone to land on.

Find the minimum possible total cost incurred before the frog reaches Stone N.

Ex → $n = 5, k = 3$

$[10, 30, 40, 50, 20]$

ans → 30

$i \rightarrow (i+1)$

$i \rightarrow (i+2)$

Ex → $n = 3 \quad k = 1$

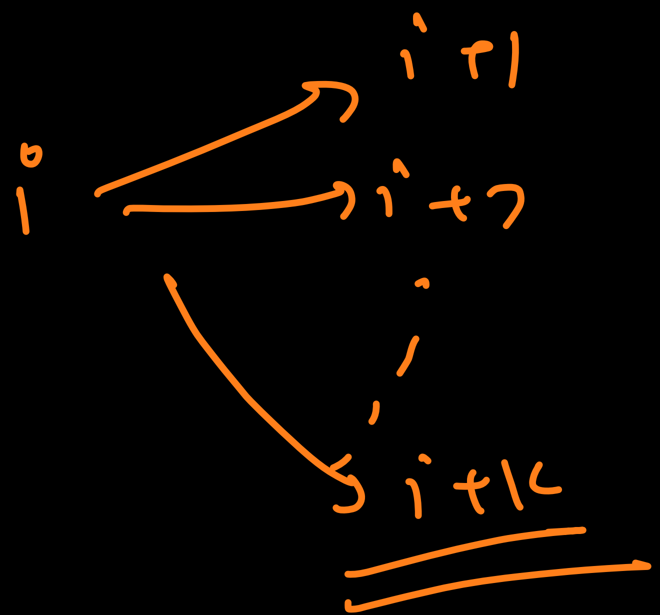
$[10, 20, 10]$

ans → 20

$$i \to i+1, \quad i \to i+2, \quad i \to i+3$$

$$i \to i+1, \quad i \to i+2, \quad i \to i+3, \quad i \to i+4$$

$$
\begin{aligned}
f(i,n) = \min\Big(& |h_i - h_{i+1}|,\ f(i+1,n), \\
& |h_i - h_{i+2}|,\ f(i+2,n), \\
& |h_i - h_{i+3}|,\ f(i+3,n) \\
\Big)
\end{aligned}
$$



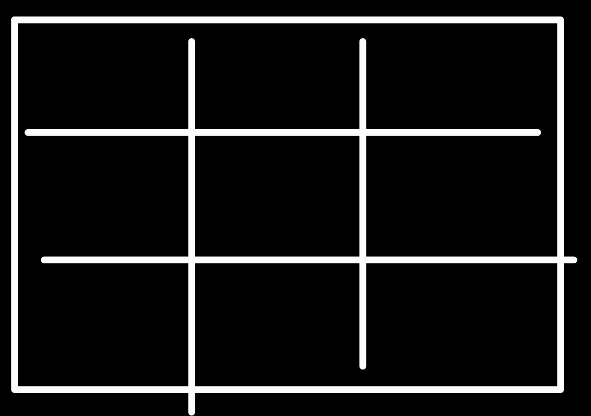
Base

$$f(i, n) = \min\left(|h_i - h_{i+j}| + f(i+j, n)\right)$$

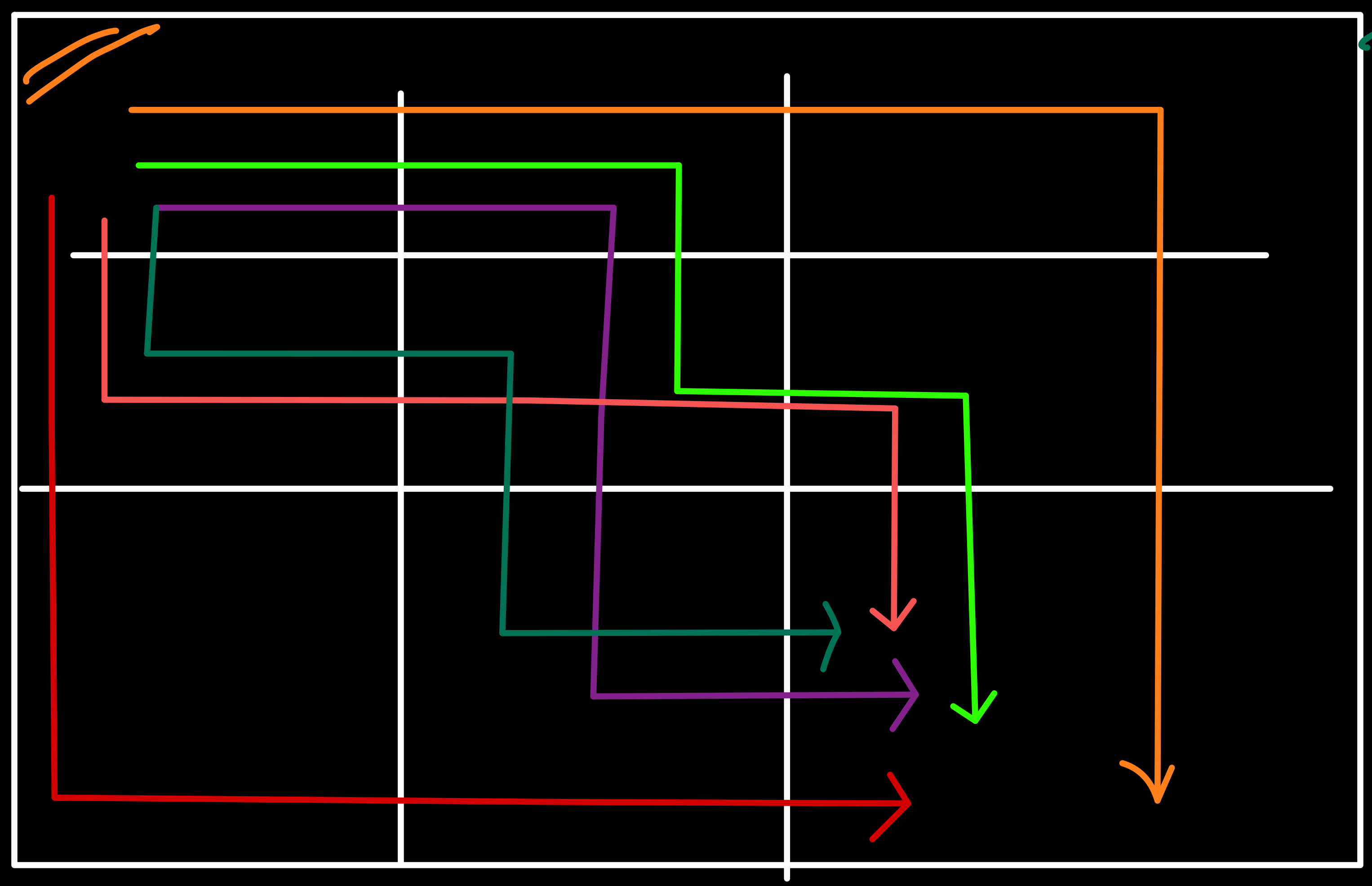
$\forall j \in [1, k]$

```
let result = Infinity
for(j=1; j<=k; j++) {
    result = min( result, f(i+j, n) + |h[i] -h[i+j]])
```

Q ⇒ Let's say, you are standing on the top left corner of a grid having dimension $n \times m$. Find the total no. of ways in which you can reach the bottom right given the fact that from any cell of the grid you can move one step down or one step right.

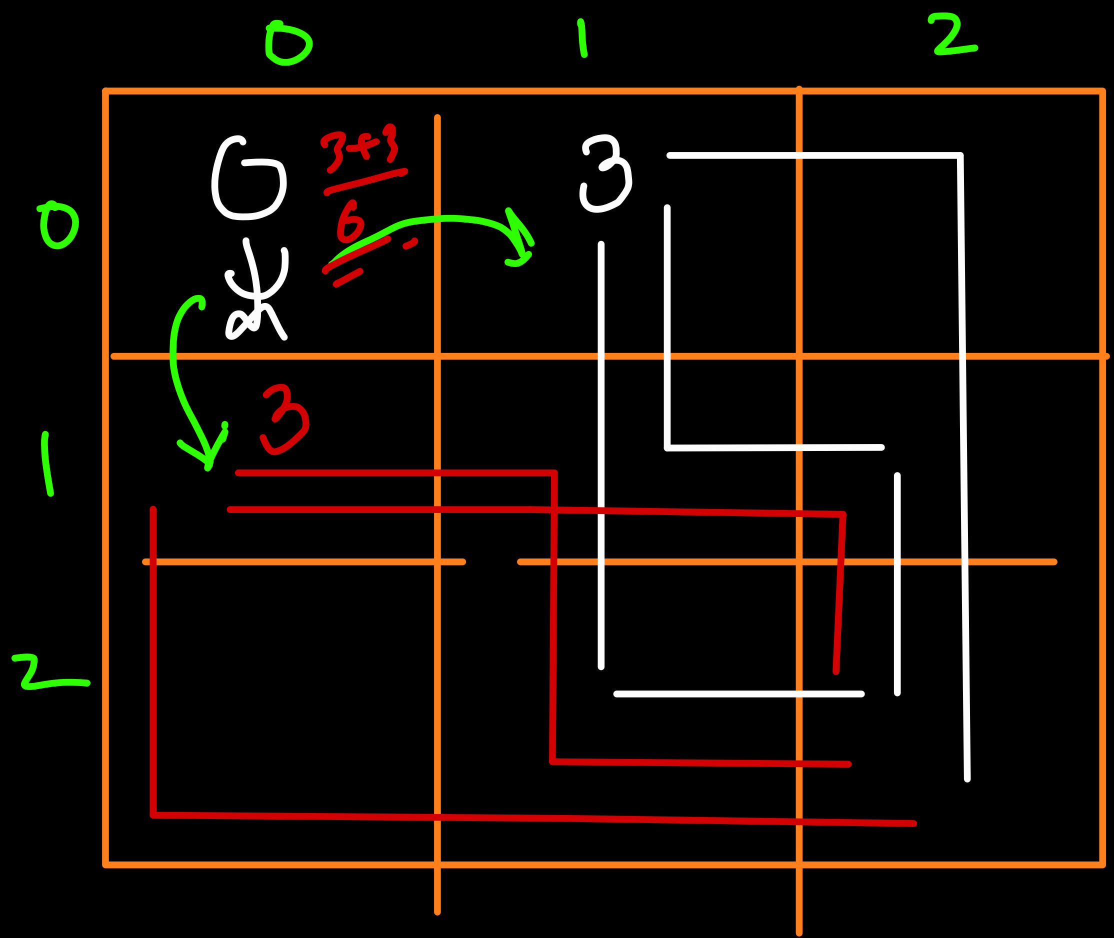


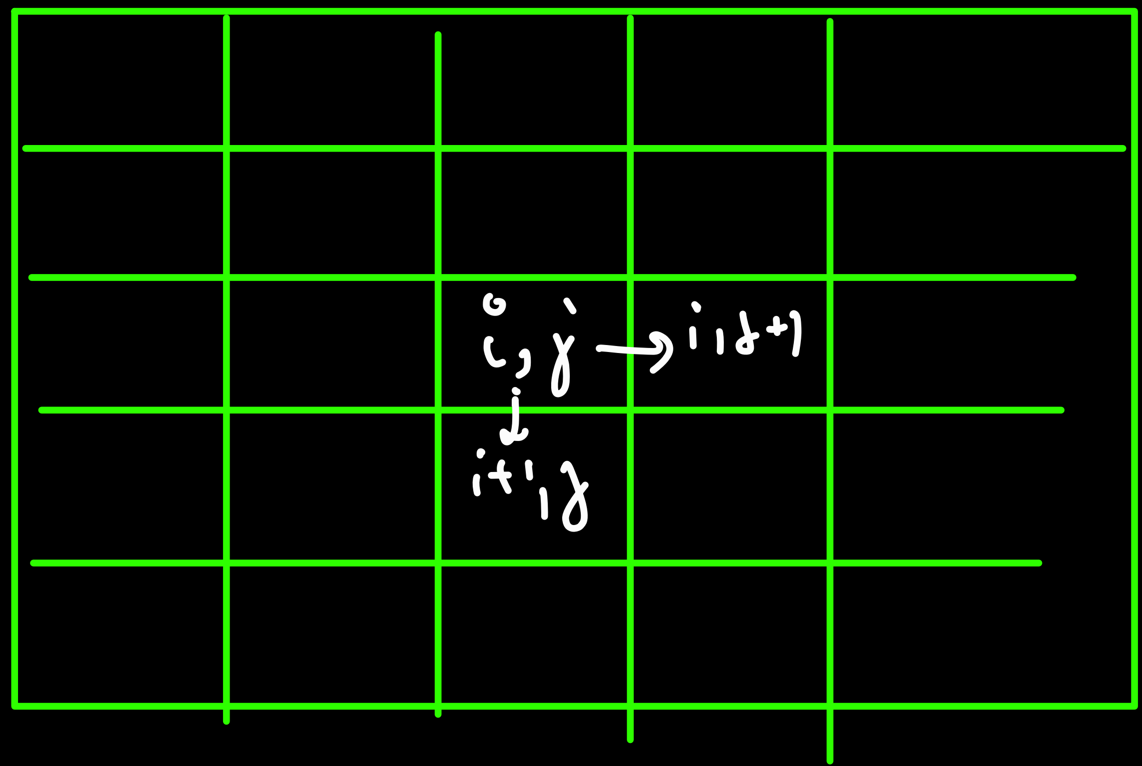
3×3 ⟶ 6



6

3x3





nxm

$f(i, j, n, m) = f(i, j+1, n, m) + f(i+1, j, n, m)$

self work

no. of ways to reach n-1, m-1 from any cell i, j

assumption

if (i == n-1 and j == m-1)
return 1;
if (i >= n or j >= m)
return 0;

actual ans → $f(0, 0, n, m)$

JOIN THE DARKSIDE

```javascript
function f(i, j, n, m) {
    if(i == n-1 && j == m-1) return 1;
    if(i >= n || j >= m) return 0;

    return f(i, j+1, n, m) + f(i+1, j, n, m);
}

console.log(f(0, 0, 2,2));
```