$$f(n, h) = \min \begin{cases} f(n-1, h) + |h_{n-1} - h_n| \\ f(n-2, h) + |h_{n-2} - h_n| \end{cases}$$

min cost to reach the n^{th} stone from i^{th} stone

$1 \rightarrow n$ → 1 based index

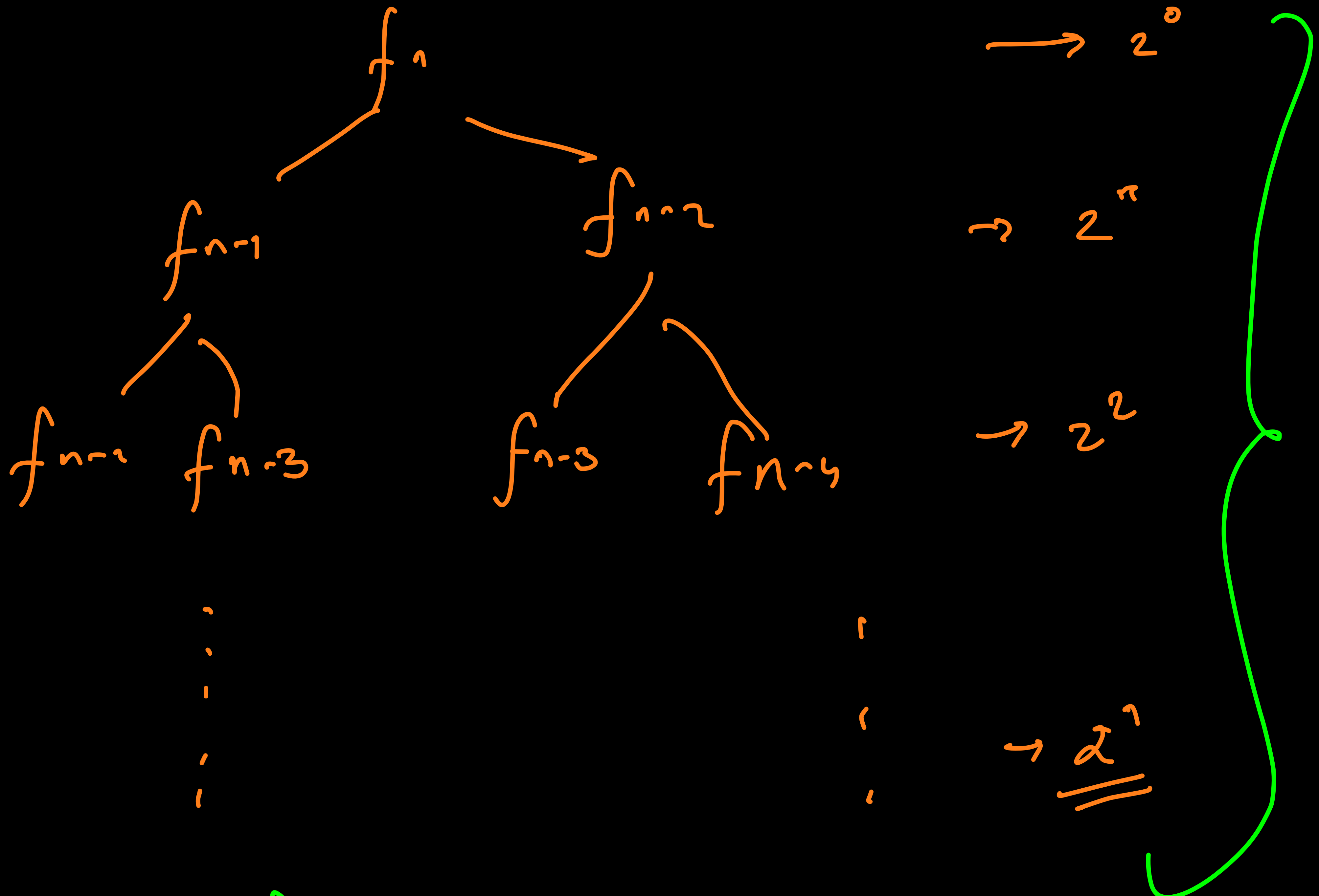
assume →
- $f(n-1, h)$ → min cost to reach $n-1$ from 1
- $f(n-2, h)$ → min cost to reach $(n-2)$ from 1

f^n works properly

_ _ _ _ _ $n-2$ $n-1$ n

~~$f_n = f_{n-1} + f_{n-2}$~~

$N \le 10^5$



$O(2^n)$

DP

memoization

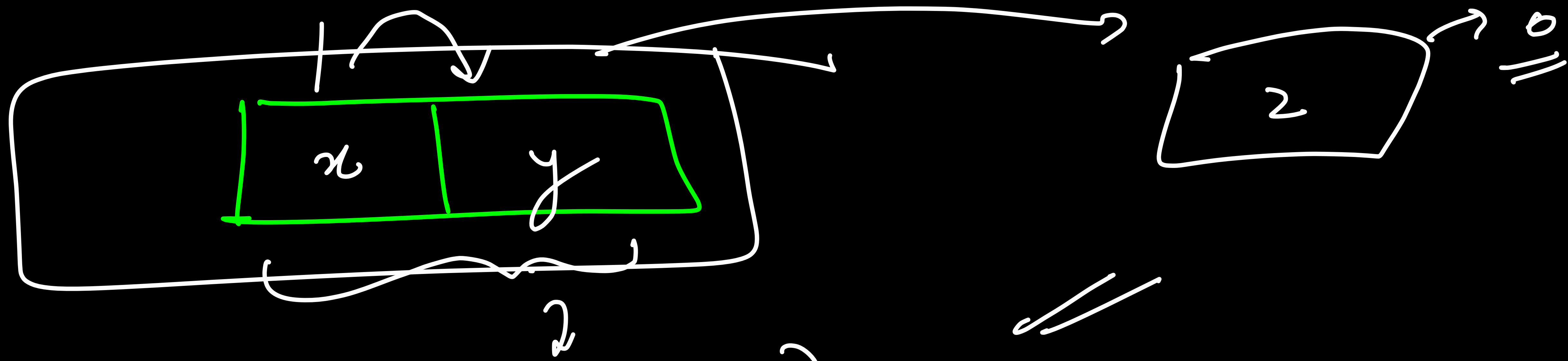
$$f(n, h) = \min \begin{cases} f(n-1, h) + |h_{n-1} - h_n| \\ f(n-2, h) + |h_{n-2} - h_n| \end{cases}$$

min cost to reach the n^{th} stone from i^{th} stone

To uniquely identify a state of a subproblem, we just need one parameter n

→ 1D

$(n \to n-1 \to n-2 \dots 1)$ $O(n)$



$ans(h_{n-1} - h_n)$

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

$$dp(i) = \min\left(dp(i-1) + \underline{|h_{i-1} - h_i|} \;,\; dp(i-2) + |h_{i-2} - h_i| \right)$$

0	1	2	3	4	5	6
//	30	10	60	10	60	50

0	1	2	3	4	5	6
//	0	20	30	20	30	40

$dp[1] = 0$

$dp[2] = abs(h[2] - h[1])$

$\underline{i = 6}$

$30 + 10 \rightarrow 40$

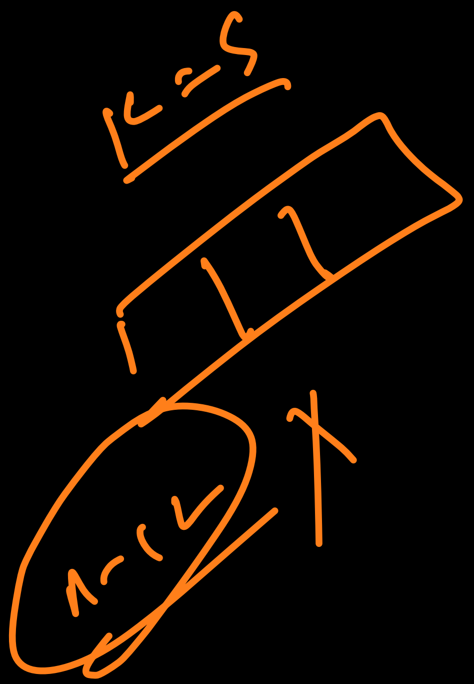
$20 + 40 \rightarrow 60$

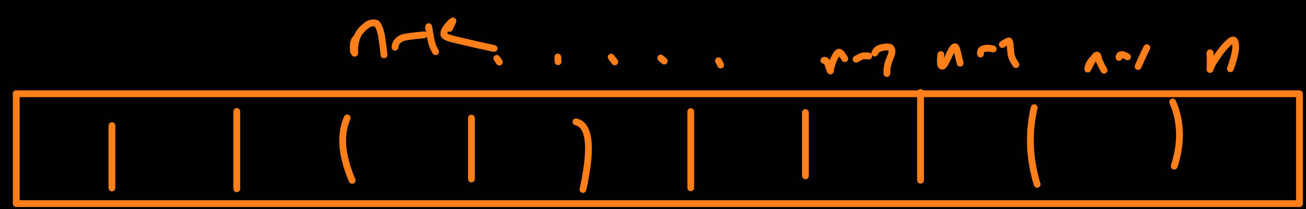
for (i = 3 ; i <= n ; i++)

ans $\underline{dp[n]}$

$$
f(n, h) = \underset{\text{min}}{=} \begin{cases} f(n-1, h) + |h_{n-1} - h_n| & \text{if } (n-1 \geq 0) \\ f(n-2, h) + |h_{n-2} - h_n| & \text{if } (n-2 \geq 0) \\ f(n-3, h) + |h_{n-3} - h_n| & \text{if } (n-3 \geq 0) \\ f(n-4, h) + |h_{n-4} - h_n| & \vdots \\ \vdots & \\ f(n-k, h) + |h_{n-k} - h_n| & \text{if } (n-k \geq 0) \end{cases}
$$

min cost to reach the n^{th} stone from i^{th} stone





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

$\forall\, j \in [1,k]$ and $(n-j \geq 0)$

↳ BU

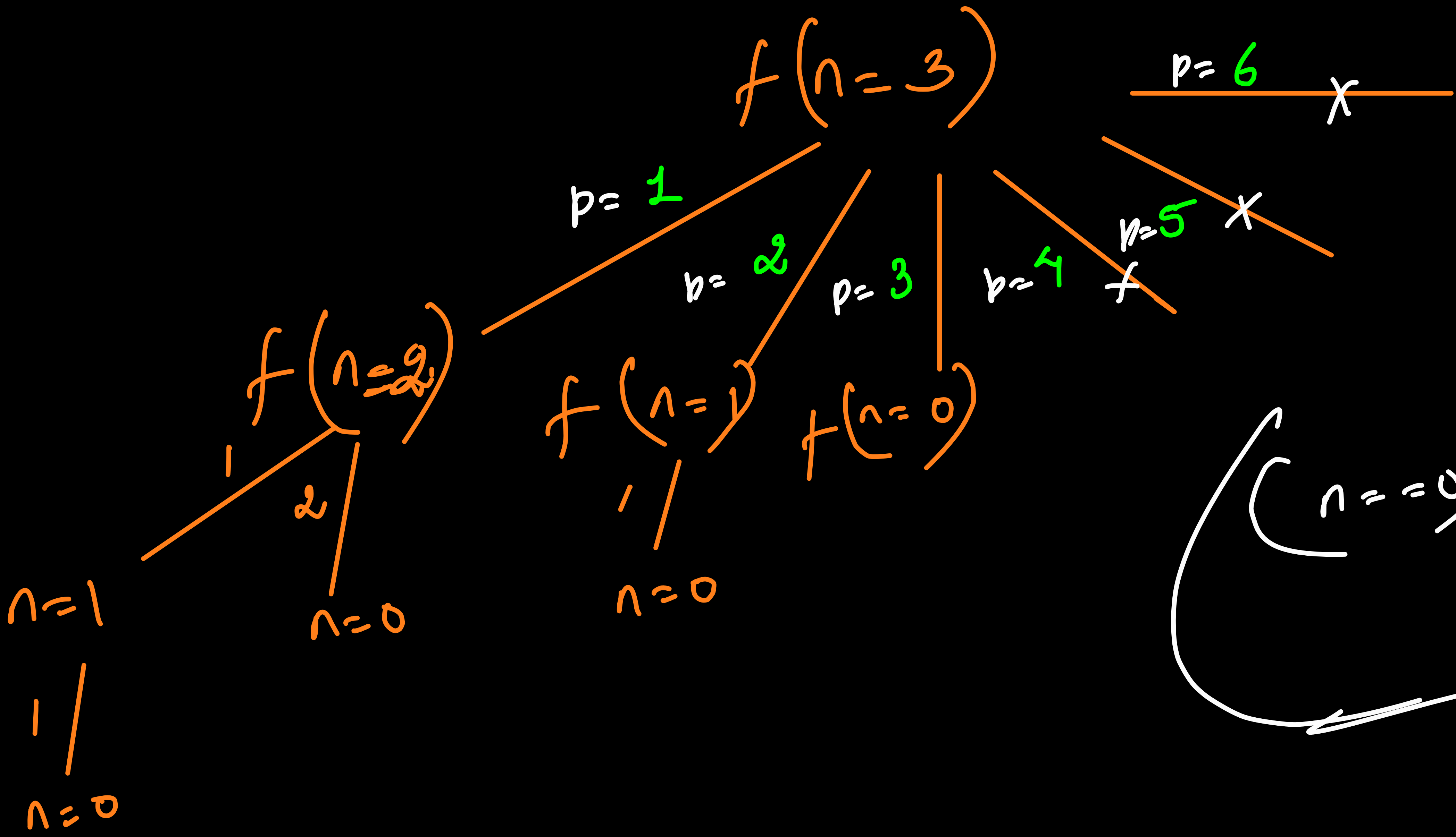
$dp[i] = \min\left(dp(i-j) + |h_{i-j} - h_n|\right) \quad \forall\, j \in [1,k],\ i-j \geq 0$

dp[n]

```
for (i=2; i<=n; i++) {
    for (j=1 ; j<= k ; j++) {
        if (i - j < 0 ) break;
        dp[i] = min (dp[i] , dp[i -j] + abs (h[i]-h[i-j]
    }
}
```

frog → minimization

Dice → convexity



$(n - p < 0)$ ✗

$(n == 0) \rightarrow$ 1 possible way

$$f(n) = f(n-1) + f(n-2) + f(n-3) + f(n-4) + f(n-5) + f(n-6)$$

$f(n)$ → no. of ways u get sum n, by dice throws

$\cancel{f(n)}$

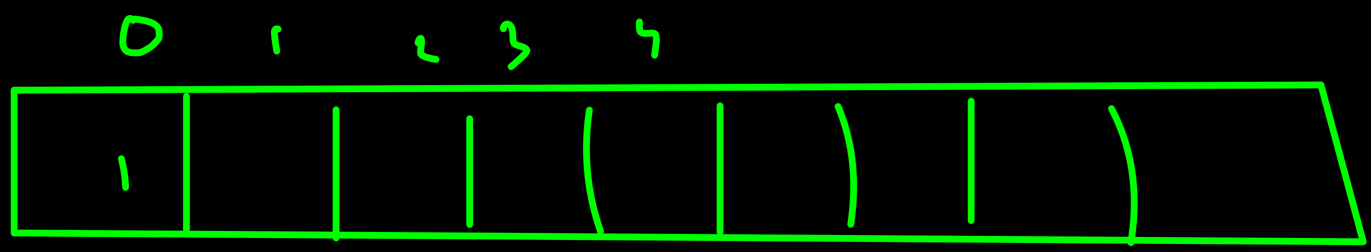
Base → $n == 0 \rightarrow 1$

$$f(n) = \sum_{j=1}^{6} f(n-j) \qquad (n-j \geq 0)$$

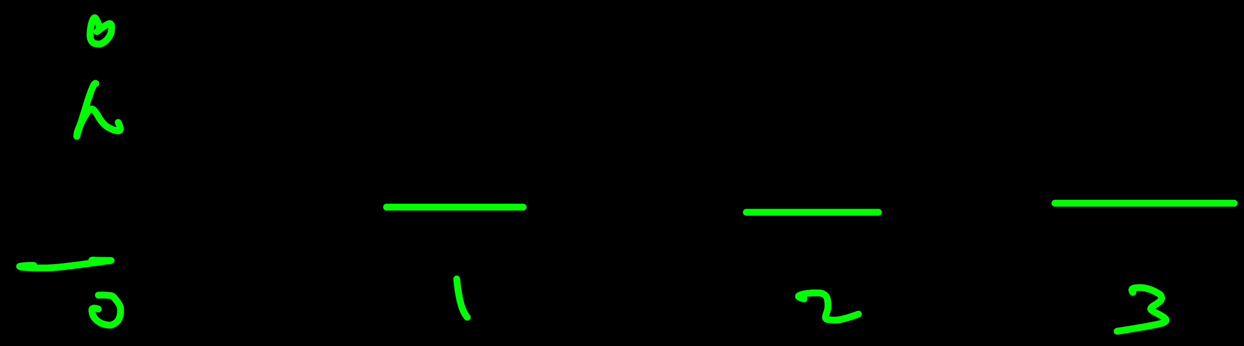
$$(a+b)\,\%\,c = (a\,\%\,c + b\,\%\,c)\,\%\,c$$

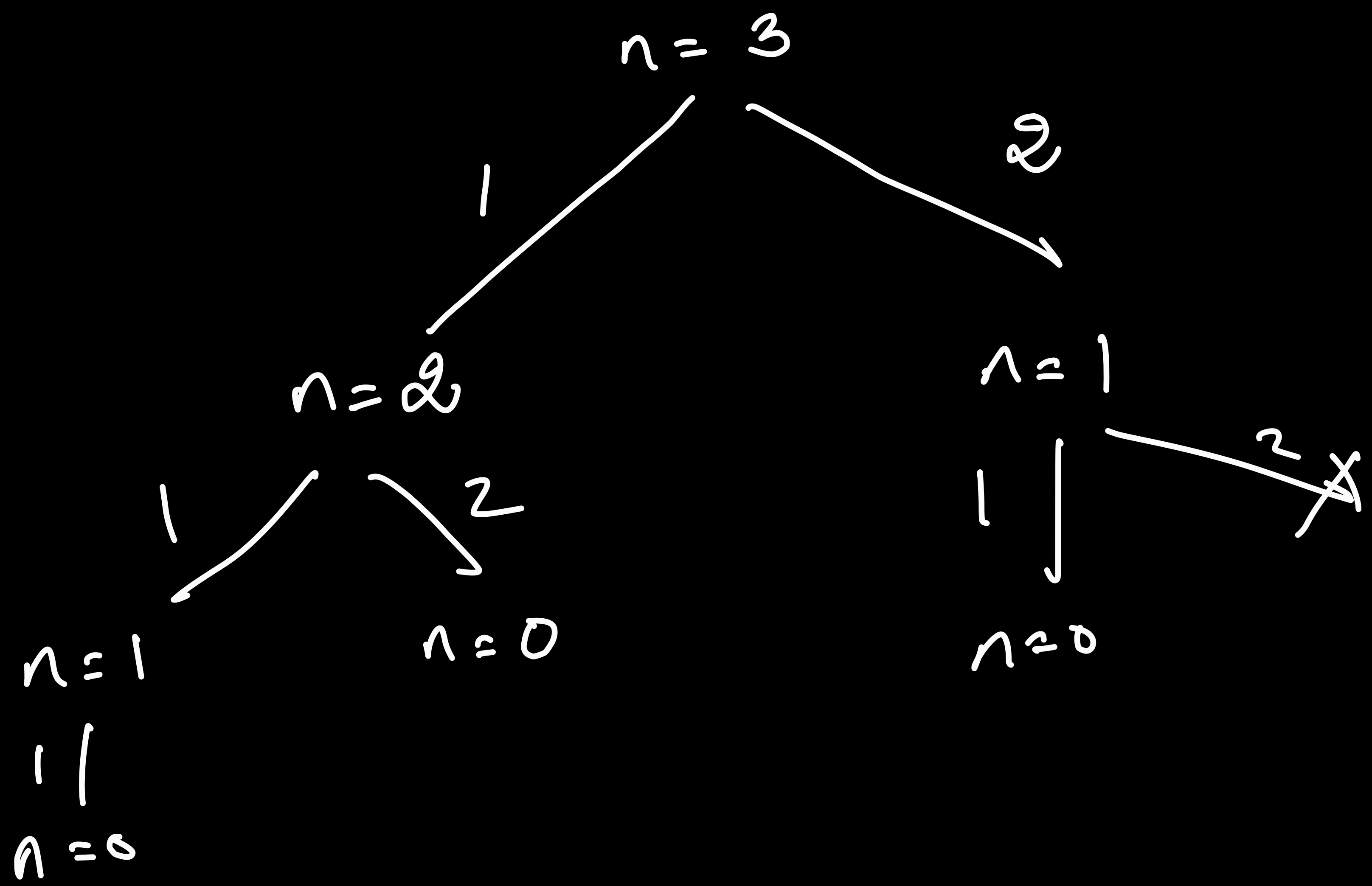
$$(a-b)\,\%\,c = (a\,\%\,c - b\,\%\,c + c)\,\%\,c$$

$$(a \times b)\,\%\,c = (a\,\%\,c \times b\,\%\,c)\,\%\,c$$

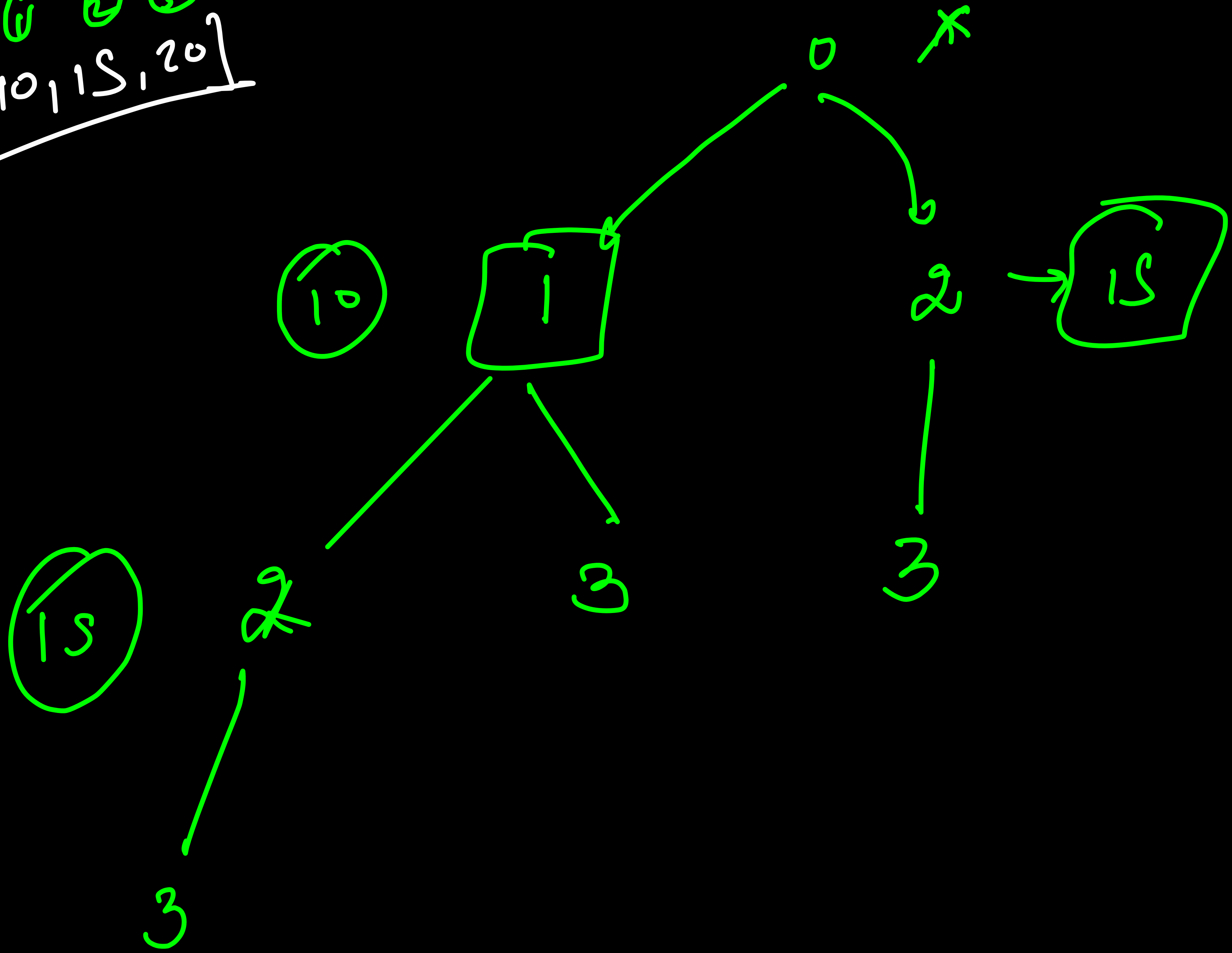



```
dp[0] = 1

for (i=1 ; i<=n ; i++) {
    for (j=1; j<=6; j++) {
        if (i-j < 0) break
        dp[i] = (dp[i] % m + dp[i-j] % m) % m
    }
}

→ dp[n]
```


0
n
0
1
2
3

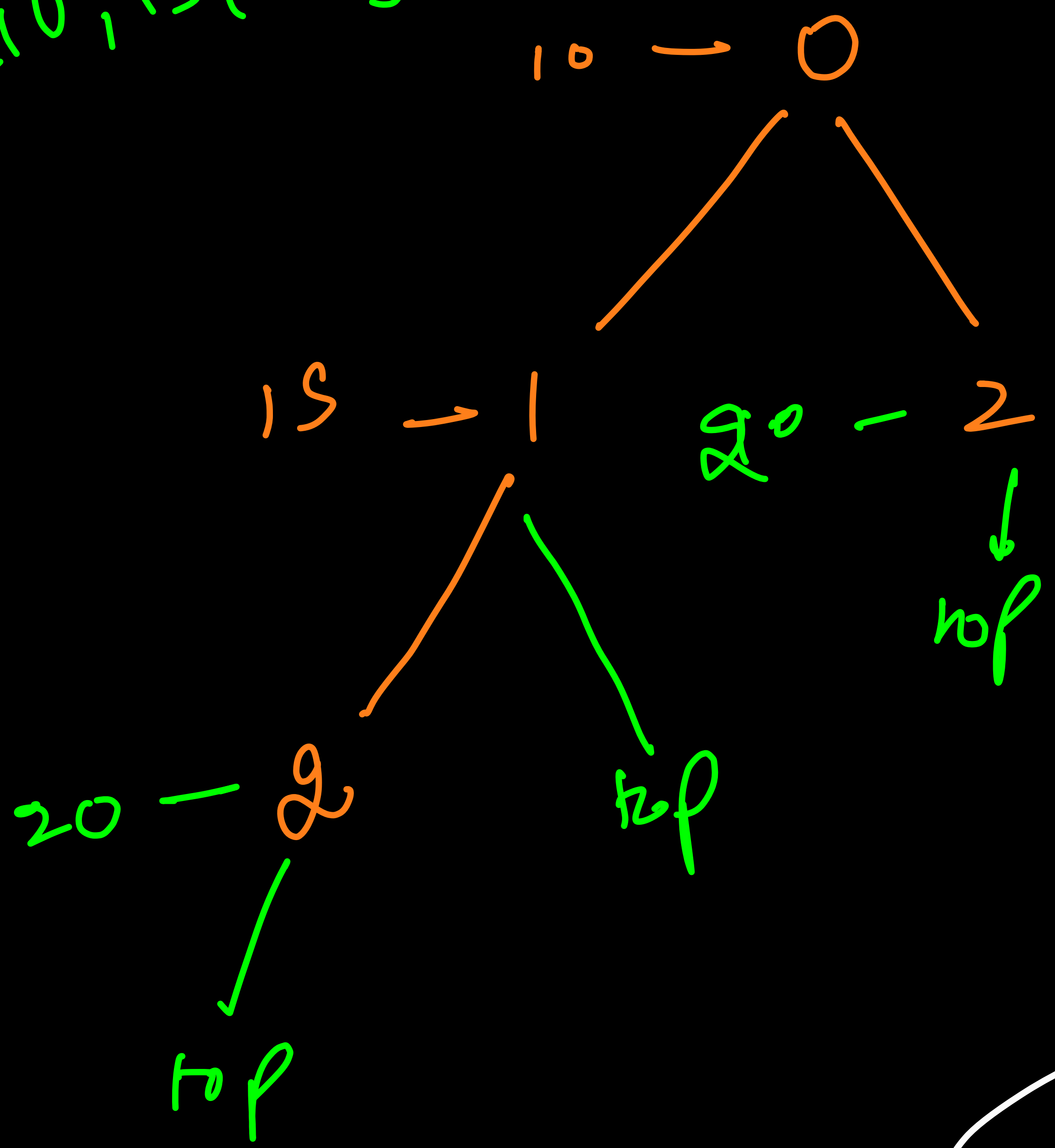

n = 3
1
2
n = 2
n = 1
1
2
1
2
n = 1
n = 0
n = 0
1
n = 0

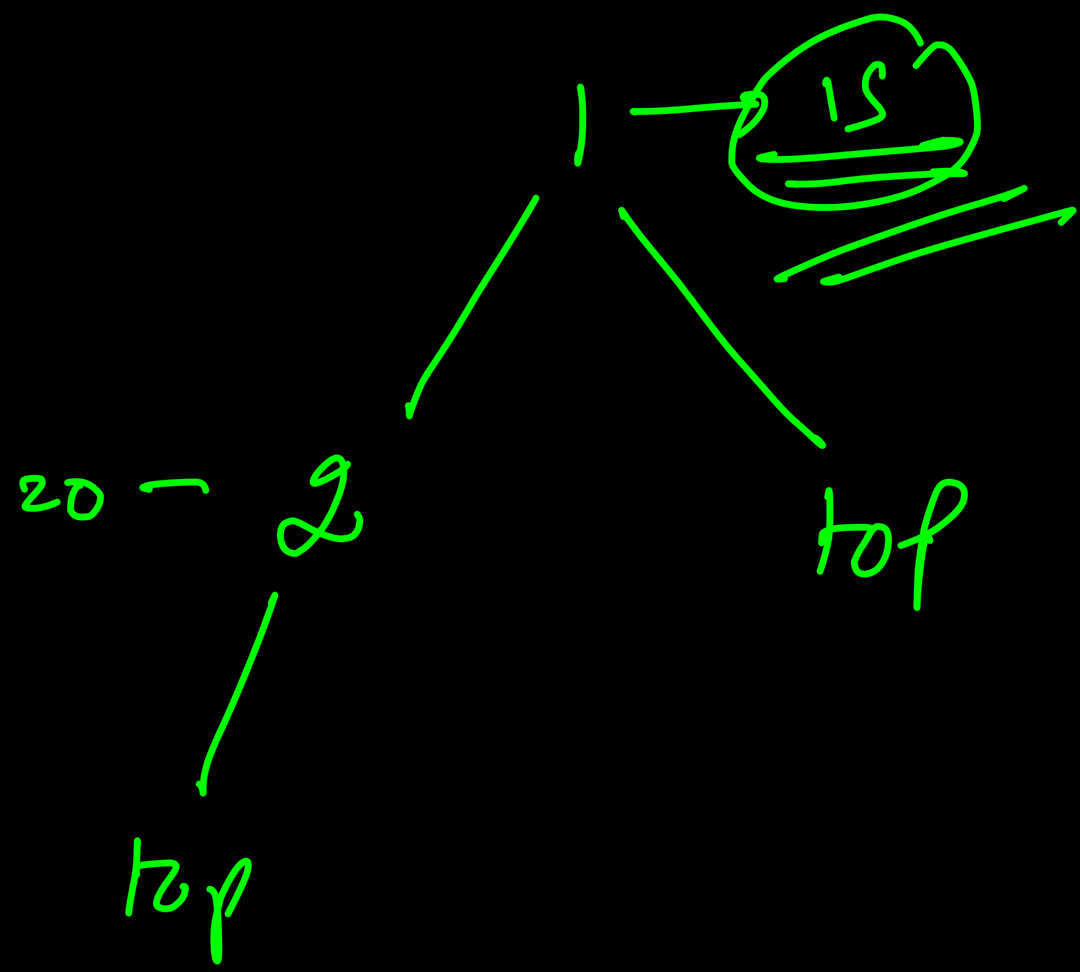
1 2 3
[10, 15, 20]



0 1 2 3

[10, 15, 20] $\boxed{\text{top}}$





$i \rightarrow i+1$

$i \rightarrow i+2$