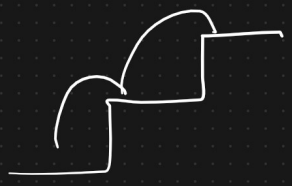
Count number of ways to reach upstairs

↳ 1 step at a time or 2 steps at a time

$n = 1$ ⟶ 1 way

$n = 2$ ⟶ 2 way

↳ (1,1), (2)



4

0 1 2 3 5 8 13 21 34

0 1 2 3 4 5 6 7 8

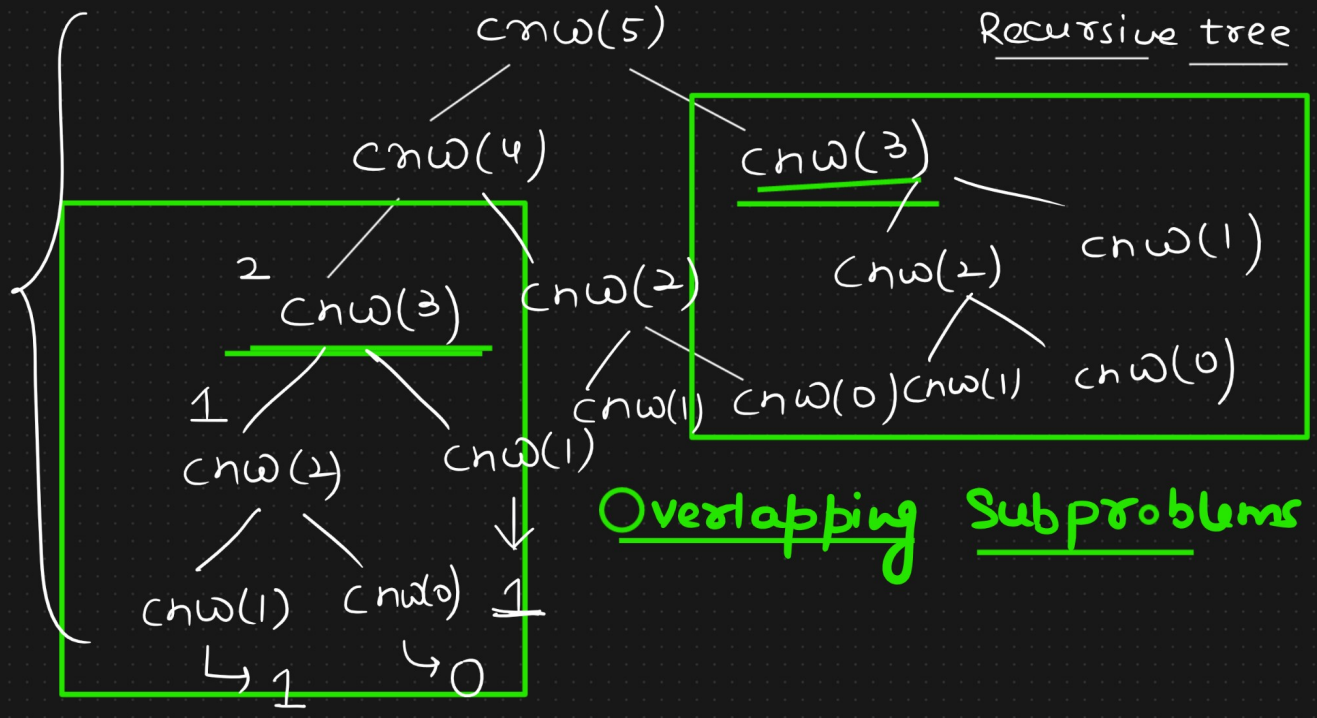
0 1 1 2 3 5 8

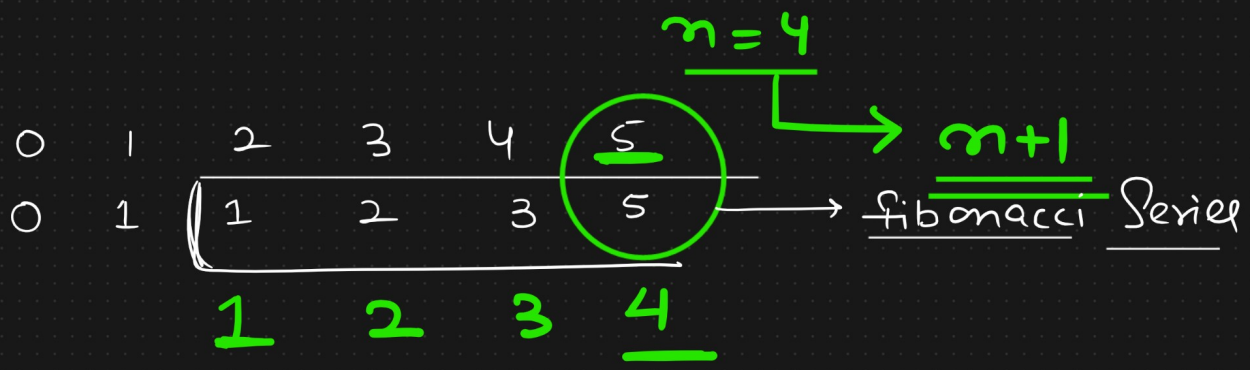
$n = 3$ ⟶ (1,2), (1,1,1), (2,1) — 3 ways

(3+1)

$n = 4$ ⟶ (1,1,1,1), (2,2), (1,2,1), (2,1,1), (1,1,2) — 5 ways

```
cnw(n)
{
    if  n<=1
        ↳ return n
    else
        cnw(n-1) + cnw(n-2)
}
```





Time complexity

CBT → Complete Binary Tree

$2^n - 1$

$O(2^n)$ → Exponential Time complexity

$n \longrightarrow$ very very large

↳ Dynamic Programming