

Count number of ways to reach upstairs

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

When we are having a single stair then we can reach upstairs only in 1 way (by climbing 1 stair at a time)

$n=1$ → 1 way

4

When we are having a two stair then we can reach upstairs in 2 ways (by climbing 1 stair at a time and by climbing 2 stairs at a time)

$n=2$ → 2 way
↳ one one step, Two step a time
 $(1, 1), (2)$



From here we can clearly observe that our current problem statement is the extension of Fibonacci series since number ways to climb the stairs forms the series that appears Fibonacci

0	1	2	3	5	8	13	21	34	→ Number of ways to climb the stairs
0	1	2	3	4	5	6	7	8	→ Number of stairs
0	1	1	2	3	5	8			→ Fibonacci series

$n=3$ → $(1, 2)$

$(3+1)$
 $(1, 1, 1)$
 $(2, 1)$ } 3 ways

When number of stairs is 3 then there can be total 3 ways

$n=4$

$(1, 1, 1, 1)$

$(2, 2)$

$(1, 2, 1)$

$(2, 1, 1)$

$(1, 1, 2)$

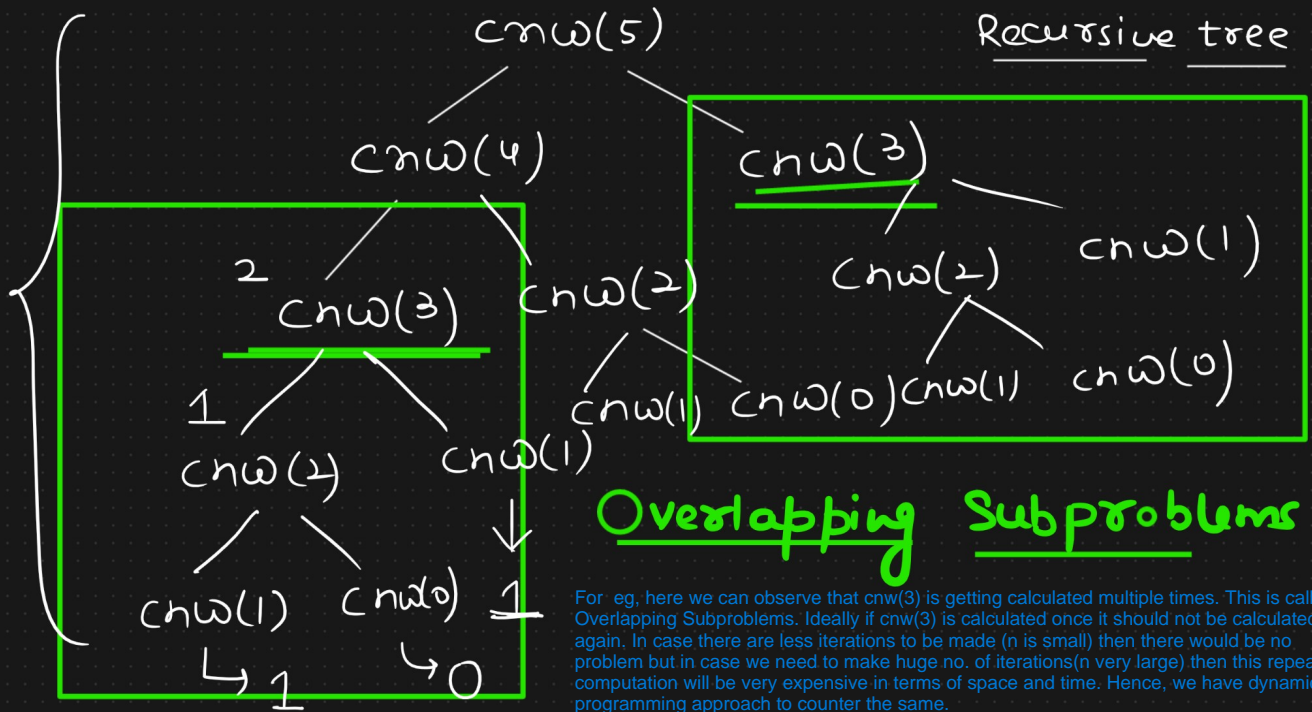
5 ways

When number of stairs is 4 then there can be total 5 ways to reach the upstairs

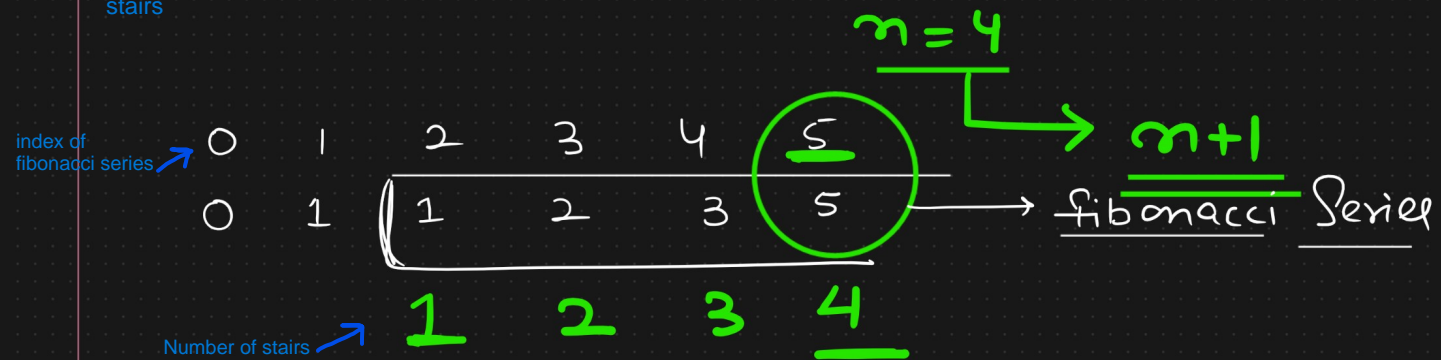
$cnw(n)$

cnw stands for count number of ways

{ if $n \leq 1$
↳ return n
else
 $cnw(n-1) + cnw(n-2)$



If we wish to climb 4 stairs ($n=4$) then there could be total 5 ways to do that. If we compare this with the Fibonacci series then we will observe that the same is represented by the $n+1$ th index that is 5th index. This is the reason why during implementation we are calling helper method as $helper(s+1)$ instructing it to return $n+1$ th index value of Fibonacci series that will represent the number of ways to climb for n number of stairs



n \rightarrow very very large
 \searrow Dynamic Programming

Since time complexity is exponential in nature hence, in case we are taking n very large say $n = 10000000000$ then we will get Kernel died popup while execution which is because of very high time complexity due to exponential execution nature. To counter this we need a more optimized approach which is Dynamic Programming. This justifies the need and importance of Dynamic programming