

## Fibonacci Pattern



Write a function to calculate the nth Fibonacci number.

Fibonacci numbers are a series of numbers in which each number is the sum of the two preceding numbers.

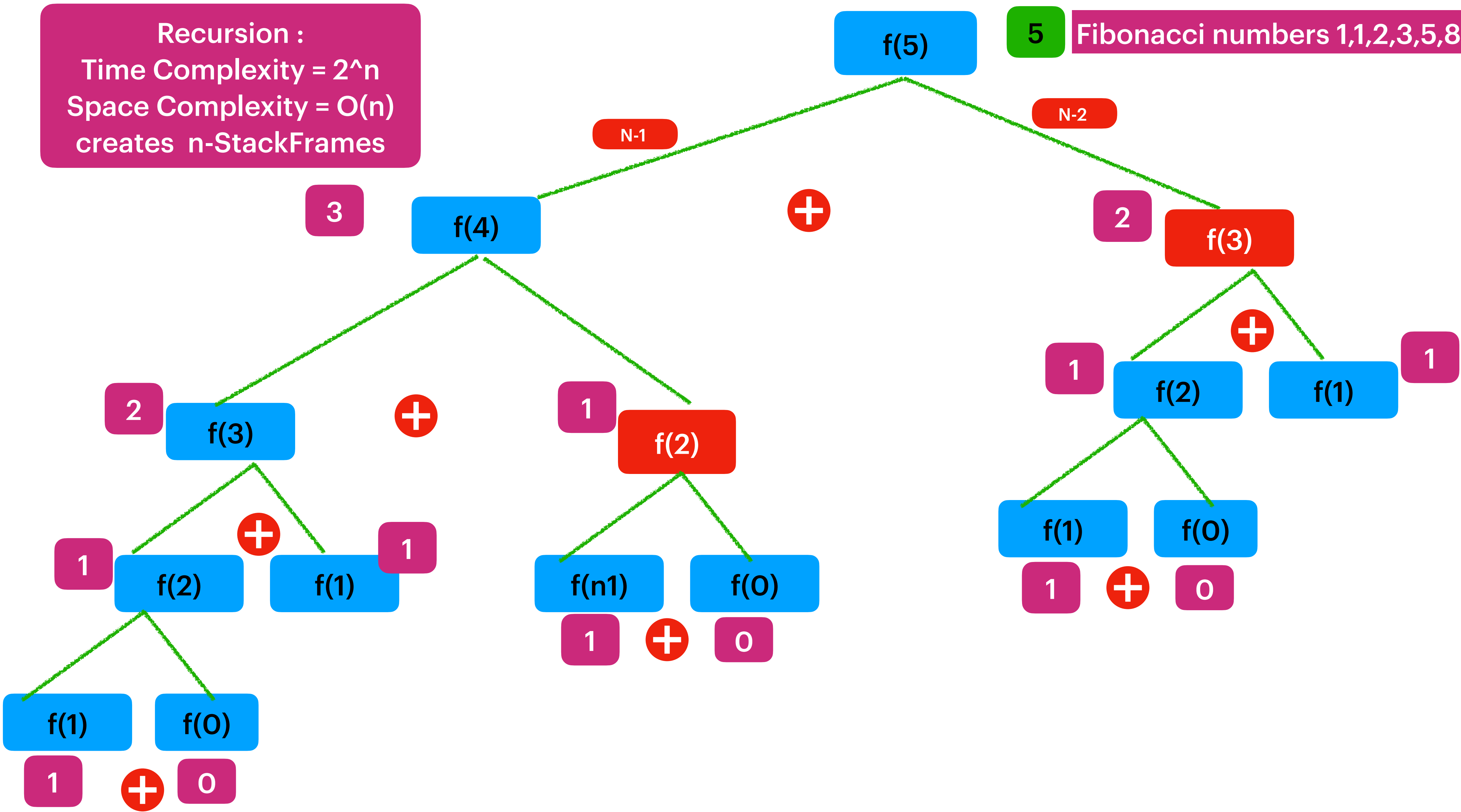
First few Fibonacci numbers are: 0, 1, 1, 2, 3, 5, 8 .....

$$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2), \text{ for } n > 1$$

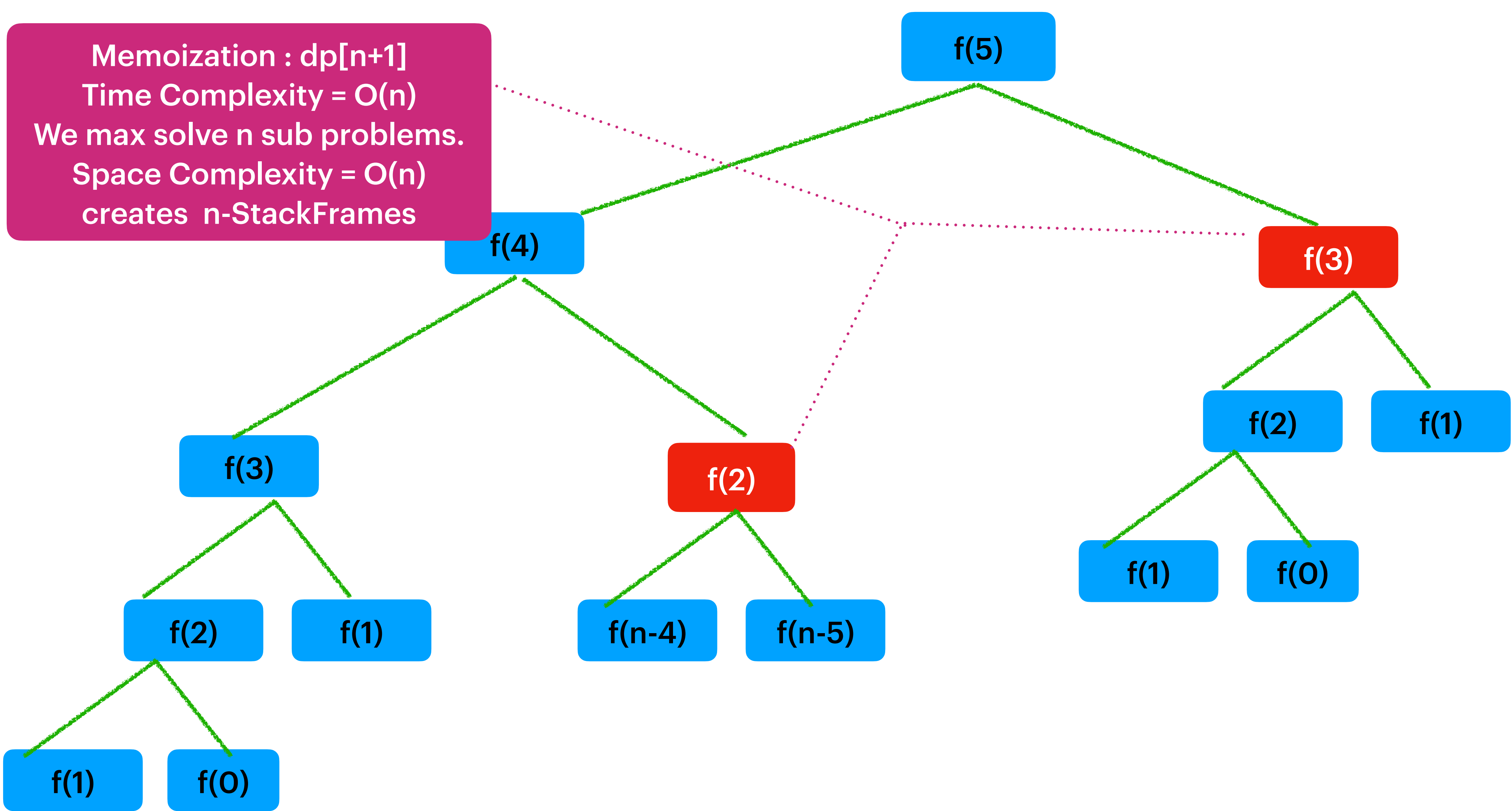
Given that:  $\text{Fib}(0) = 0$ , and  $\text{Fib}(1) = 1$

Recursion :  
Time Complexity =  $2^n$   
Space Complexity =  $O(n)$   
creates n-StackFrames

5 Fibonacci numbers 1,1,2,3,5,8,13,21 ...



Memoization :  $dp[n+1]$   
Time Complexity =  $O(n)$   
We max solve  $n$  sub problems.  
Space Complexity =  $O(n)$   
creates  $n$ -StackFrames



Tabulation  $dp[n+1]$

$dp[0] = 0$

$dp[1] = 1$

$dp[2] = dp[1] + dp[0] = 0 + 1 = 1$

$dp[n] = dp[n-1] + dp[n-2];$