Fibonacci Numbers

(Algorithmic Problems)

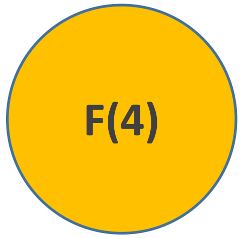
Fibonacci Numbers

$$F(N) = F(N-1) + F(N-2)$$

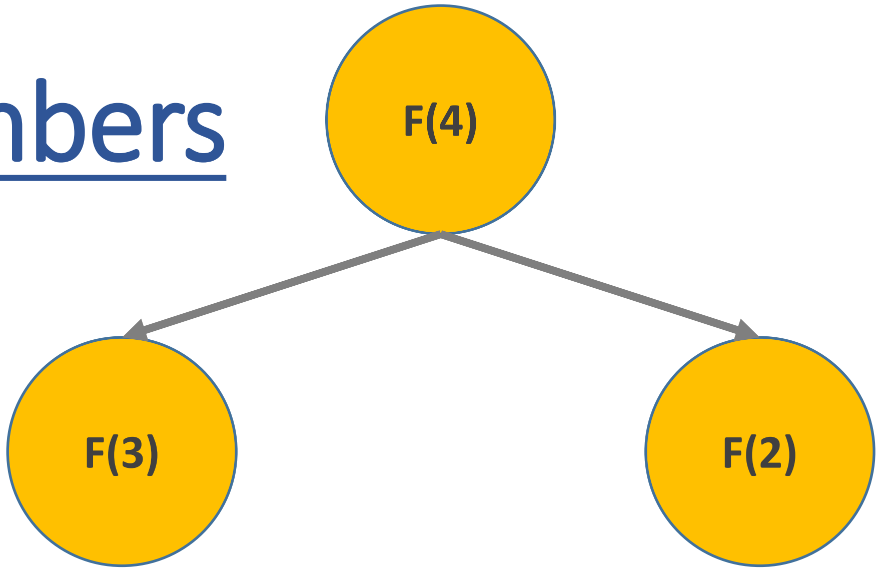
$$F(0) = 0 \quad F(1) = 1$$

What is the problem with the recursive formula?
we keep calculating same subproblems
(Fibonacci numbers) over and over again

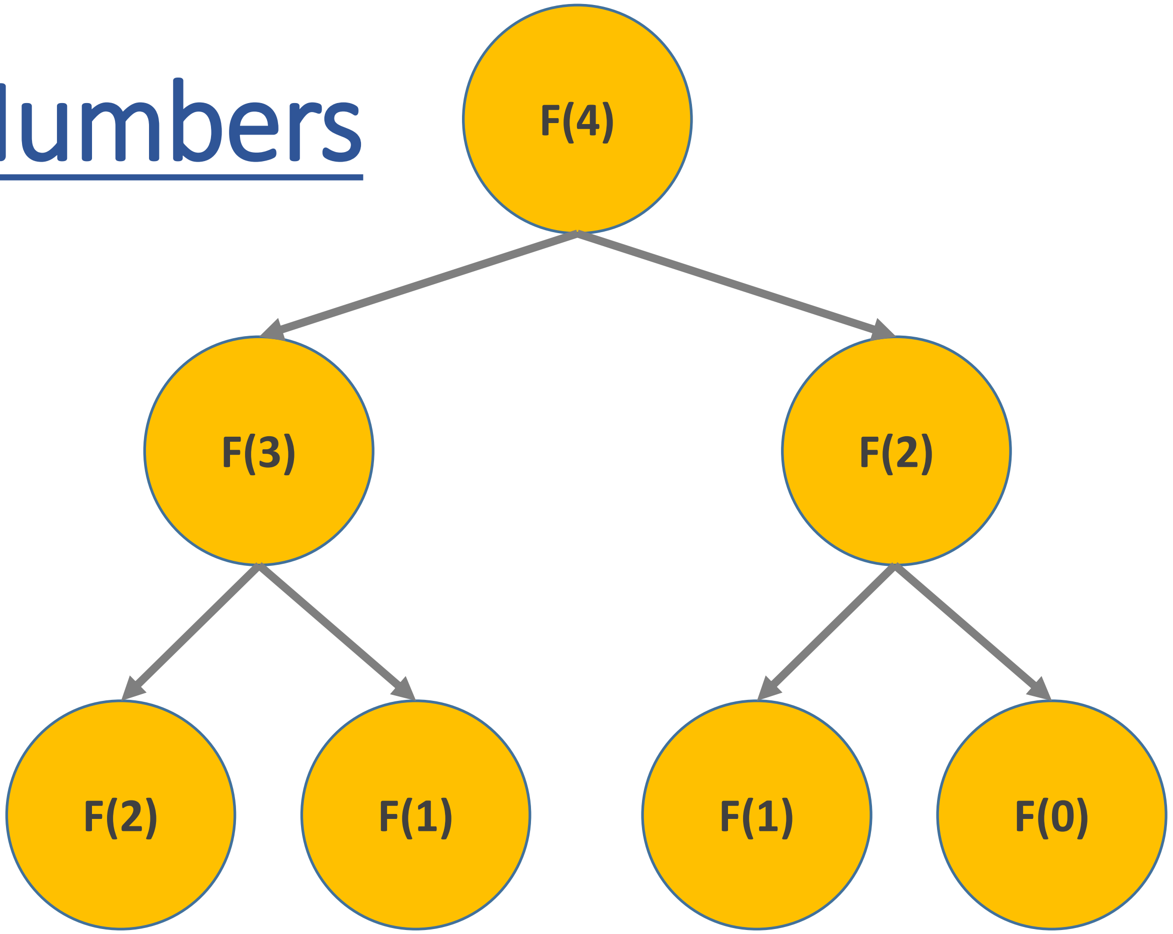
Fibonacci Numbers



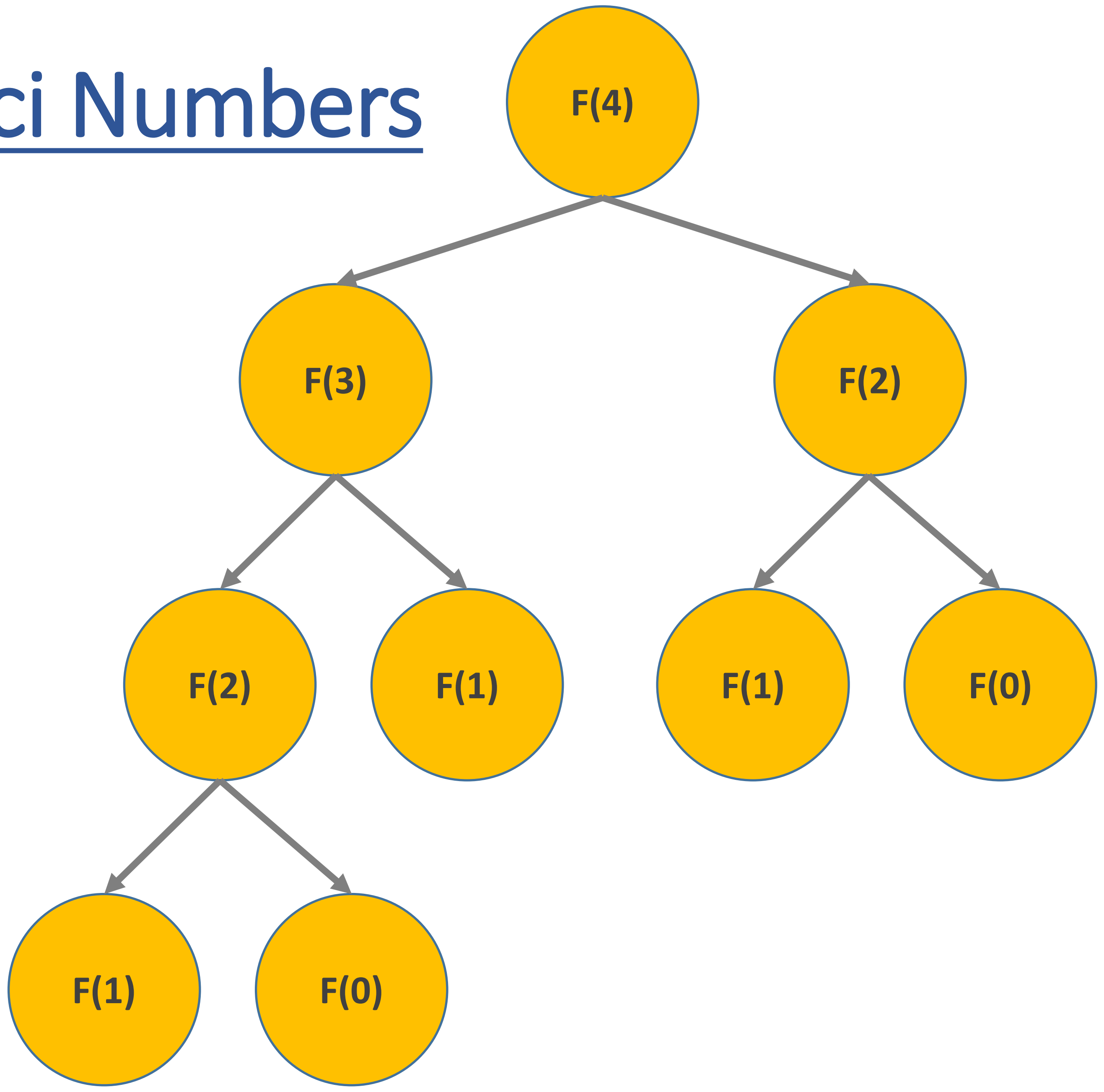
Fibonacci Numbers



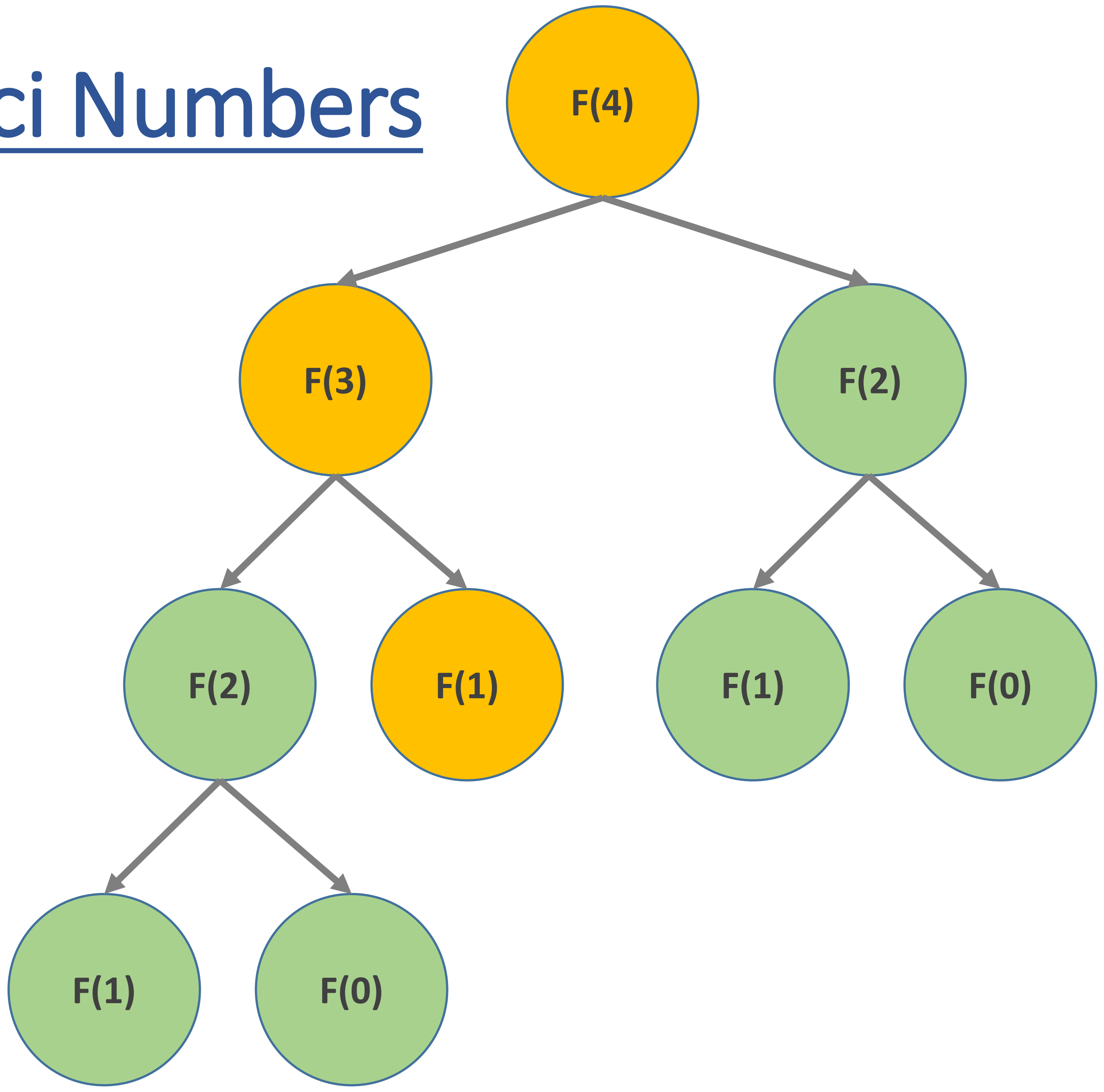
Fibonacci Numbers



Fibonacci Numbers



Fibonacci Numbers



Fibonacci Numbers

- let's use dynamic programming and **memoization** in order to avoid recalculating a subproblem over and over again
- we should use an associative array abstract data type to store the solution for the subproblems - **O(1)** time complexity
- on every **f()** method call - we insert the calculated value if necessary
- instead of the **$O(2^N)$** exponential time complexity we will have **O(N)** time complexity + requires **O(N)** space