Recursive Fibonacci number calculation

Recall that recursive calculation of the n^{th} Fibonacci number:

- We store the two base cases, $F(1) = F(2) = 1$, in an array $Fib[1...n]$ (i.e., $Fib[1] = 1, Fib[2] = 1$).
- For $i = 3$ up to n, we compute $Fib[i]$ using the rule:
  - $Fib[i] = Fib[i-1] + Fib[i-2]$
- Return Fib[n]

The running time here is easy to analyze: there are n entries, and each one uses at most C operations for some constant C.
Hence the total time is at most Cn time.