CSCI 332, Fall 2025

In-class Activity

Recall the definition of the nth Fibonacci number:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

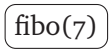
1. Write a recursive algorithm to compute $F(n)$. Check your answer by implementing it in Python here: https://www.online-python.com/vpEjKaXOBt. (Link is also in Discord.)

   fibo($n \in \mathbb{Z}^{\geq 0}$):

2. What is the recurrence for the runtime of your algorithm? The base cases are filled in for you. Remember that the recurrence should be the sum of the time for the recursive calls and the time for the non-recursive work.

   $T(0) = 1$, $T(1) = 1$, $T(n) =$ _____ for $n > 1$.

3. Draw the recursion tree for fibo(7). Label each node with the function call that is being made. The first node is filled in for you.



4. What is your guess for the asymptotic number of nodes in the tree, in terms of the input number n?

   num. nodes is $\Theta($_____$)$

5. Do you see any way to improve the runtime of the algorithm?