Project 1

Theoretical Analysis

We have two algorithms to find the nth Fibonacci number, F_n, one recursive and one iterative. We proceed with the theoretical analysis of these two algorithms, and find the order of growth of each.

Recursive algorithm

We have the following recursive definition for F_n:

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

To find the general form for $F(n)$, we solve a linear homogenous recurrence relation, and have

$$F(n) = \frac{1}{\sqrt{5}}(\varphi)^n - \frac{1}{\sqrt{5}}(\hat{\varphi})^n$$

with

$$\varphi = \left(\frac{1+\sqrt{5}}{2}\right) \text{ and } \hat{\varphi} = \left(\frac{1-\sqrt{5}}{2}\right)$$

To find the order of growth of $F(n)$, we need to calculate the limit as n approaches infinity. Since $|\hat{\varphi}| \approx 0.62 < 1$, then

$$\lim_{n\to\infty} \frac{1}{\sqrt{5}}(\hat{\varphi})^n = 0.$$

And since, $|\varphi| \approx 1.62 > 1$, then

$$\lim_{n\to\infty} \frac{1}{\sqrt{5}}(\varphi)^n = \infty.$$

So,

$$\lim_{n\to\infty} F(n) = \frac{1}{\sqrt{5}} \lim_{n\to\infty} (\varphi)^n + (\hat{\varphi})^n = \frac{1}{\sqrt{5}} \lim_{n\to\infty} (\varphi)^n = \infty.$$

Thus, $F(n) \in \Theta(\varphi^n)$ or $F(n) \in \mathcal{O}(2^n)$

Iterative algorithm

We have the following pseudocode of the iterative algorithm from the book on page 82:

ALGORITHM $Fib(n)$

//Computes the nth Fibonacci number iteratively by using its definition
//Input: A nonnegative integer n
//Output: The nth Fibonacci number
$F[0] \leftarrow 0, F[1] \leftarrow 1$
for $i \leftarrow 2$ **to** n **do**
    $F[i] \leftarrow F[i-1] + F[i-2]$
end for
return $F[n]$

The first two assignments to $F[0]$ and $F[1]$ are of constant time, so the efficiency of the algorithm is dependent on the **for** loop. Since the operation that occurs on every iteration of the loop is addition, which can be completed in constant time, we have

$$C(n) = \sum_{i=2}^{n} 1 = (n - 2 + 1) = n - 1$$

So $C(n) \in \Theta(n)$

Empirical Analysis