Rose Whitt
2/4/22
COMP 322

Homework 1 Written

1.1.1 Recursive Fibonacci Complexity

Given the Fibonacci formula:

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

We know that the total work when n <= 1 is:

$$Work(fib(n <= 1)) = O(1)$$

Since the recursive algorithm terminates upon reaching fib(0) or fib(1):

$$Work(fib(n)) = Work(fib(n - 1)) + Work(fib(n - 2)) + O(1)$$

Given that the big-O of *fib(n - 1)* is:

$$Work(fib(n - 2)) = O(2^{n-1})$$

We can translate our previous Work formula into Big-O and reduce it to the upper bound:

$$Work(fib(n)) = 2^{n-1} + 2^{n-2} + 1 = O(2^{n})$$

Considering the closed form for a Fibonacci number, we can use the golden ratio (ϕ) provided and the above formulas to conclude that the complexity of the recursive fibonacci algorithm is:

$$\boldsymbol{Work(fib(n)) = O(\phi^{n})}$$

1.1.2 Memoized Fibonacci Complexity

(Part 1)

Because memoization stores the values as they are calculated and calls upon them instead of recalculating the same value (like the 1.1.1 algorithm), we can safely assume that the big-O formula for the WORK will be less than that of the non-memoized recursive algorithm found above.

Given that the *memoized* lookup takes *O(n)* time, we can calculate the big-O formula for the WORK as function of *n* performed by a call to *fib(n)* (where *n < MaxMemo*) for the very first time:

$$\begin{aligned} Work(fib(n)) = \quad & Work(fib(1)) + Work(fib(0)) + Work(Addition) \\ + \quad & Work(fib(1)) + Work(addition) \\ + \quad & Work(Lookup\ of\ fib(2)) + Work(Addition) \\ + \quad & \ldots \\ + \quad & Work(Lookup\ of\ fib(n - 2)) + Work(Addition) \end{aligned}$$

Using our previously know Work and Lookup values, we can reduce this:

$$\begin{aligned} Work(fib(n)) &= (1 + 1 + 1) + (1 + 1) + (2 + 1) + \ldots + (n - 2 + 1) \\ &= (2 + (1 + 2 + 3 + \ldots + (n - 2)) \\ &= 2 + \frac{(n-2)(n-1)}{2} \\ &= \boldsymbol{O(n^2)} \end{aligned}$$

Rose Whitt
2/4/22
COMP 322
(Part 2)
Let us say that the value of *n* is *k* greater than the previously called number. The work calculation would then be:

Work(fib(n)) = Work(Lookup of fib(n - k - 1)) + Work(Lookup of fib(n - k)) + Work(Addition)
+ *Work(Lookup of fib(n - k + 1)) + Work(Addition)*
+ . . .
+ *Work(Lookup of fib(n - 2)) + Work(Addition)*

Reducing this:

Work(fib(n)) = ((n - k - 1) + (n - k) + 1) + ((n - k + 1) + 1) + . . . + ((n - 2) + 1)
= -1 + ((n - k) + (n - k + 1) + (n - k + 2) + . . . + (n - 1))
$= -1 + \frac{n(n-1)}{2} - \frac{(n-k-1)(n-k)}{2}$
$= 1 + \frac{n^2 - n - n^2 + 2nk + k^2 - n - k}{2}$
= O(2nk - 2n - k)
= O(nk)