CS 161 **Pre-Lecture Exercise for Lecture 12**
Winter 2019 **Date:** February 25, 2019

Pre-lecture exercises will not be collected for credit. However, you will get more out of each lecture if you do them, and they will be referenced during lecture. We recommend **writing out** your answers to pre-lecture exercises before class. Pre-lecture exercises usually should not take you more than 30 minutes.

Consider the Fibonacci numbers, defined by

$$F(0) = F(1) = 1$$

and

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

For example, the first several Fibonacci numbers are:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...$$

Consider the following divide-and-conquer algorithm to compute Fibonacci numbers.

```
def Fibonacci(n):
    if n == 0 or n == 1:
        return 1
    return Fibonacci(n-1) + Fibonacci(n-2)
```

1. Is this algorithm correct?
2. What is the running time of this algorithm? You don't need to find it exactly, but is it $O(n)$? $O(n^2)$? $O(n^3)$? $O(n^c)$ for any constant c?
3. How could you make this algorithm better?