Tutorial 5, Week 6 - Divide and Conquer and Recursion vs Iteration

1 Before Your Tutorial

- Refresh your memory about the Fibonacci sequence:

$$f_n = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f_{n-1} + f_{n-2} & \text{otherwise} \end{cases}$$

  i.e. The first two Fibonacci numbers are both 1, and every other Fibonacci number is the sum of the previous two.

- Relax and enjoy the long weekend.

2 During Your Tutorial

- Implement a Divide and Conquer algorithm that computes the Fibonacci sequence. The recursive definition given above is a pretty big hint about how to do this. (*i.e* implement a function with a parameter n that computes the n^{th} Fibonacci number.)
- Find the largest n you can compute f_n for using your Divide and Conquer algorithm (don't wait too long, just get a feel for how far you can go).
- Implement an iterative algorithm to compute the n^{th} Fibonacci number.
- See (at least roughly) what the highest n is now (for which you can compute f_n).
- Estimate (or prove if you can) the big-oh running time for both implementations. Check this experimentally by counting steps for different values of n, then plotting the function that they outline.

Have a look at the extension exercises and at least think about them.

3 Extensions

Without just looking them up online, can you think of any other approaches to compute f_n? How do they compare to the basic approaches?

If you can't, look some up and compare them!