## INDUCTION AND SEQUENCES

This handout is to follow up on some ideas from lecture.

The *Fibonacci sequence* is defined as follows:

$$\begin{aligned} f_0 &= 0, \\ f_1 &= 1, \\ f_n &= f_{n-1} + f_{n-2} \quad \text{for all } n \geq 2. \end{aligned}$$

This definition lets us compute the sequence iteratively. Thus it begins

$$0, \quad 1, \quad 1, \quad 2, \quad 3, \quad 5, \quad 8, \quad 13, \quad 21, \quad 34, \quad 55, \quad 89, \quad 144, \quad \cdots.$$

A closed form expression for the Fibonacci number $f_n$ involves the roots of the polynomial

$$X^2 - X - 1.$$

By the quadratic formula, the roots are

$$\tau = \frac{1+\sqrt{5}}{2}, \qquad \widetilde{\tau} = \frac{1-\sqrt{5}}{2},$$

and since these are the roots it follows that

$$\tau^2 = \tau + 1, \qquad \widetilde{\tau}^2 = \widetilde{\tau} + 1. \tag{1}$$

A bit of algebra shows that

$$f_0 = \frac{\tau^0 - \widetilde{\tau}^0}{\sqrt{5}}, \qquad f_1 = \frac{\tau^1 - \widetilde{\tau}^1}{\sqrt{5}}.$$

And a bit of algebra combines with the identities (1) to show that

$$\begin{aligned} &\textit{for any } n \geq 2, \\ &\textit{if } f_{n-1} = \frac{\tau^{n-1} - \widetilde{\tau}^{n-1}}{\sqrt{5}} \textit{ and } f_{n-2} = \frac{\tau^{n-2} - \widetilde{\tau}^{n-2}}{\sqrt{5}} \\ &\textit{then consequently } f_n = \frac{\tau^n - \widetilde{\tau}^n}{\sqrt{5}}. \end{aligned}$$

It follows by mathematical induction that

$$\textit{for all } n \geq 0, \quad f_n = \frac{\tau^n - \widetilde{\tau}^n}{\sqrt{5}}.$$

You might enjoy writing a Java program to compute Fibonacci numbers iteratively and again using the closed form expression, and compare the results. You will need to use `int` or `long` quantities for whole numbers and `float` or `double` quantities for floating-point numbers.

One comment to be made here is that the mathematical induction argument is not very satisfying: it tells us nothing about how on earth we ever found the closed form for $f_n$ in the first place. The induction only verifies that the formula, apparently given to us by oracle, works.

Another comment is that the closed form does not make it all obvious that the Fibonacci numbers are integers.

A third comment is that since $\tau > 1$ while $|\widetilde{\tau}| < 1$, an approximate formula for the $n$th Fibonacci number as $n$ gets large is

$$f_n \sim \frac{\tau^n}{\sqrt{5}},$$

and consequently the ratios of consecutive Fibonacci numbers tend to a limit,

$$\lim_{n\to\infty} \frac{f_{n+1}}{f_n} = \tau.$$

The numerically unstable sequence that we discussed in class is defined as follows:

$$\begin{aligned} a_0 &= 1, \\ a_1 &= 0.99, \\ a_n &= 100.99a_{n-1} - 99a_{n-2} \quad \text{for all } n \geq 2. \end{aligned}$$

Again the definition lets us compute the sequence iteratively.

It is clear that

$$a_0 = (0.99)^0, \qquad a_1 = (0.99)^1.$$

And a bit of algebra shows that

$$\begin{aligned} &\textit{for any } n \geq 2, \\ &\textit{if } a_{n-1} = (0.99)^{n-1} \textit{ and } a_{n-2} = (0.99)^{n-2} \\ &\textit{then consequently } a_n = (0.99)^n. \end{aligned}$$

It follows by mathematical induction that

$$\textit{for all } n \geq 0, \quad a_n = (0.99)^n.$$

As we saw in class, the Java program to compute this sequence by iteration fails catastrophically.