Worksheet 5 - Fibonacci

Oliver Ray

Week 5

This worksheet uses the Fibonacci series to explore some of the concepts introduced in the theory part of the course. You will write and compare two implementations of this series using recursion and iteration, respectively; and you will do some proof by induction[1].

Recall that the n'th number in the Fibonacci series $(0, 1, 1, 2, 3, 5, 8, 13, \dots)$ is recursively defined as follows:

$$f(n) = \begin{cases} 0 & \text{if} \quad n = 0 \\ 1 & \text{if} \quad n = 1 \\ f(n-1) + f(n-2) & \text{if} \quad n \geq 2 \end{cases}$$

1. Write a function `int f(int n)` that uses *recursion* (but no *iteration*) to compute the n'th number in the Fibonacci series. **Hint:** use a conditional to translate the above definition directly into C code.

2. Write a function `int g(int n)` that uses *iteration* (but no *recursion*) to compute the n'th number in the series. **Hint:** use a loop to update two variables representing the current and next number in the series.

3. Find the largest number in the Fibonacci series that can be computed by `f` and `g` (assuming that an `int` is represented by 4 bytes) and estimate how long each function actually takes to do so. Explain!

4. Prove (by induction) that the following holds for all $n \geq 0$:

$$f(n) = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n\sqrt{5}}$$

[1]Note that proof by induction was taught by Kerstin on Monday and we looked at an example in class yesterday. Further examples can be found in K.H. Rosen's book on *Discrete Mathematics and Its Applications*.