Exercise 2

Given the following definition of Fibonacci numbers

Let $n \in \mathbb{N}$. The n-th Fibonacci number be defined by

$$F_n = 0, \ if \ n = 0$$
$$F_n = 1, \ if \ n = 1$$
$$F_n = F_{n-1} + F_{n-2}, \ otherwise$$

The first 10 Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Part 1

Initialize an empty repository and use it for this exercise. From now on, push your changes to an accessible remote repository on ci.inf-i2.uni-jena.de.

Part 2

Implement different algorithms to compute Fibonacci numbers. You should implement the following three algorithms

1. Algorithm derived from the definition (very inefficient, but obviously correct)
2. Algorithm using linear memory in n and running in linear time in n.
3. Algorithm using constant memory and running in linear time in n.

Obviously every iteration is an improvement over the previous one. Use git according to the standards established in the lecture. Consider your code to be potentially used as a library by someone else. Use the branching feature to implement every version of the algorithm separately, and merge it back into master only after completion.
Try to deliberately create a merge conflict.

Optionally: work in groups of two, if you can. Both should implement the 1st solution as a start. From there, choose which repository to use from now on. Have the other partner clone the chosen repository. Let the cloning partner be user y, the other be user x. Have user y also create an additional repository on the ci server and add this repository as a second(!) remote, besides origin (the one cloned from).
Likewise, have user x add user y's newly created repository as a remote as well. From there: work as a team. Have one partner implement version 2, the other one version 3 in a separate branch. Create a pull request when done. Review the other one's code before merging.

Part 3

Implement a basic time measurement and evaluate the solutions for sufficiently large *n*.