Fibonacci

October 2022

- Why have you chosen to implement your solution as you did?

The recursive implementation is kept as short as possible. The edge cases occurring with n = 1 and n = 0 is handled separately. For all other n's the function recursively calls on itself to calculate the Fibonacci number. The main target was to keep the performance as good as possible.

The non-recursive implementation was a bit more tricky to keep short. The goal was to keep the loop as short and understandable as possible. While handling the edge cases where a bit more complex than with the recursive implementation, a decent solution was found after some time.

- Pros and cons with the two alternatives you have found?

Recursive:

1. Pros:
   - (a) Short and understandable code
   - (b) Easy to implement
2. Cons:
   - (a) Less effective
   - (b) Hard to debug
   - (c) Each call only finds one Fibonacci number

Non-Recursive:

1. Pros:
   - (a) Best performance
   - (b) Finds whole sequences of Fibonacci numbers
2. Cons:
   - (a) Requires more code for same functionality