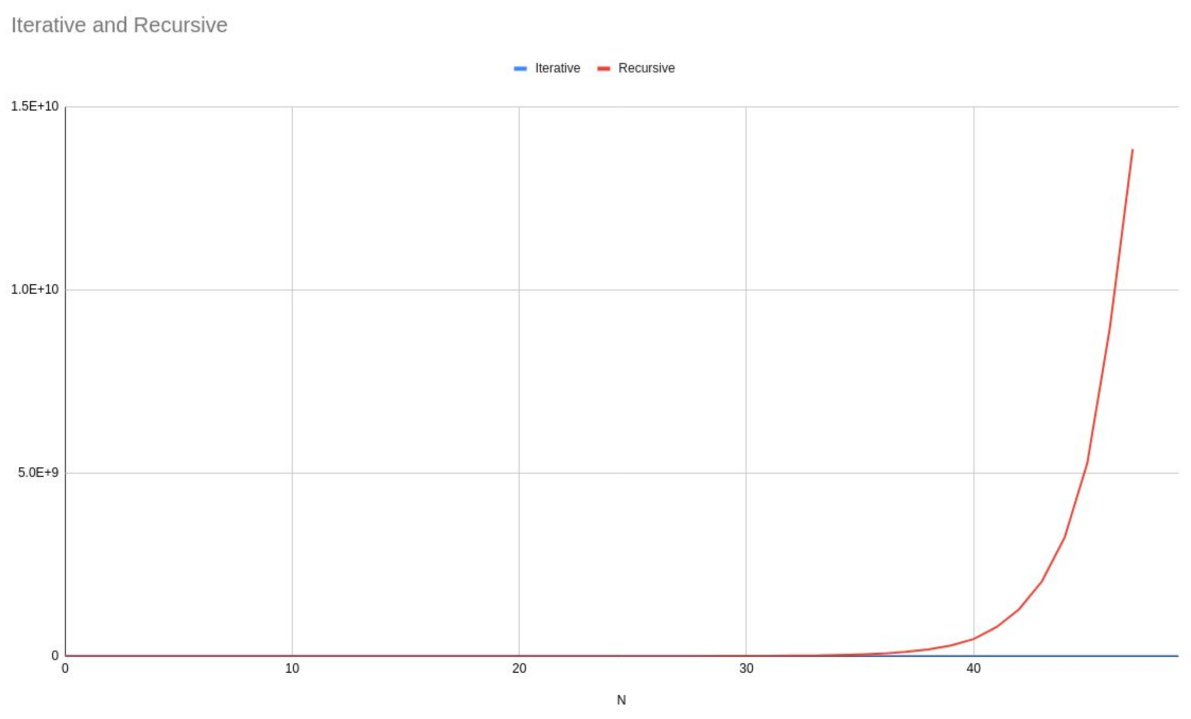
Fibonacci

Exercises

1. Below is an iterative algorithm that computes Fibonacci numbers. Write a recursive function to do the same.
    a. On github as Fibonacci.java
2. Test both algorithms with various sizes of Ns. What do you find?
    a. The recursive implementation sees its running time increase exponentially with respect to N. The iterative function increases linearly
3. What is the time complexity of both functions?
    a. Iterative: linear
    b. Recursive: exponential

Below is a graph comparing the running times for the first 50 fibonacci numbers using the iterative (blue) and recursive (red) algorithms. The iterative results are difficult to see, as they are almost parallel to the x-axis because of the difference between the iterative and recursive running times. This graph highlights the inefficiency of the recursive method for calculating fibonacci numbers.



Hanoi - The Monks need your help!

Tasks:

1. Implement Hanoi in java
2. Test with various size disks
3. Output the moves for the monks as step-by-step instructions so the monks can end the world

Result calling my Hanoi algorithm with 3 disks:

```
Move disk from source to destination
Move disk from destination to auxilliary
Move disk from auxilliary to source
Move disk from source to destination
Move disk from destination to auxilliary
Move disk from auxilliary to source
Move disk from source to destination
Move disk from destination to auxilliary
```

Result calling my Hanoi algorithm with 6 disks:

```
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
Move disk from auxilliary to destination
Move disk from destination to source
Move disk from source to auxilliary
```

These are both verifiably correct by comparing them with the known outputs for these disk numbers.