Let us say n = fibonacci(N) and m = fibonacci(N - 1)

fibonacci(N) = fibonacci(N-1) + fibonacci(N-2)

OR n = m + k where k < m.

Therefore the step

```
n = n % m will make n = k

swap(n, m) will result in

n = fibonacci(N-1)

m = k = fibonacci(N-2)
```

So, it will take N steps before m becomes 0.

This means, in the worst case, this algorithm can take N step if **n** is Nth fibonacci number.

Think of whats the relation between N and n.