Fibonacci and Euclid – first encounter.

Proposition *For any* $n \geq 1$, *Euclid's algorithm takes* $n - 1$ *trips through the loop to compute* $\gcd(f_n, f_{n-1})$.

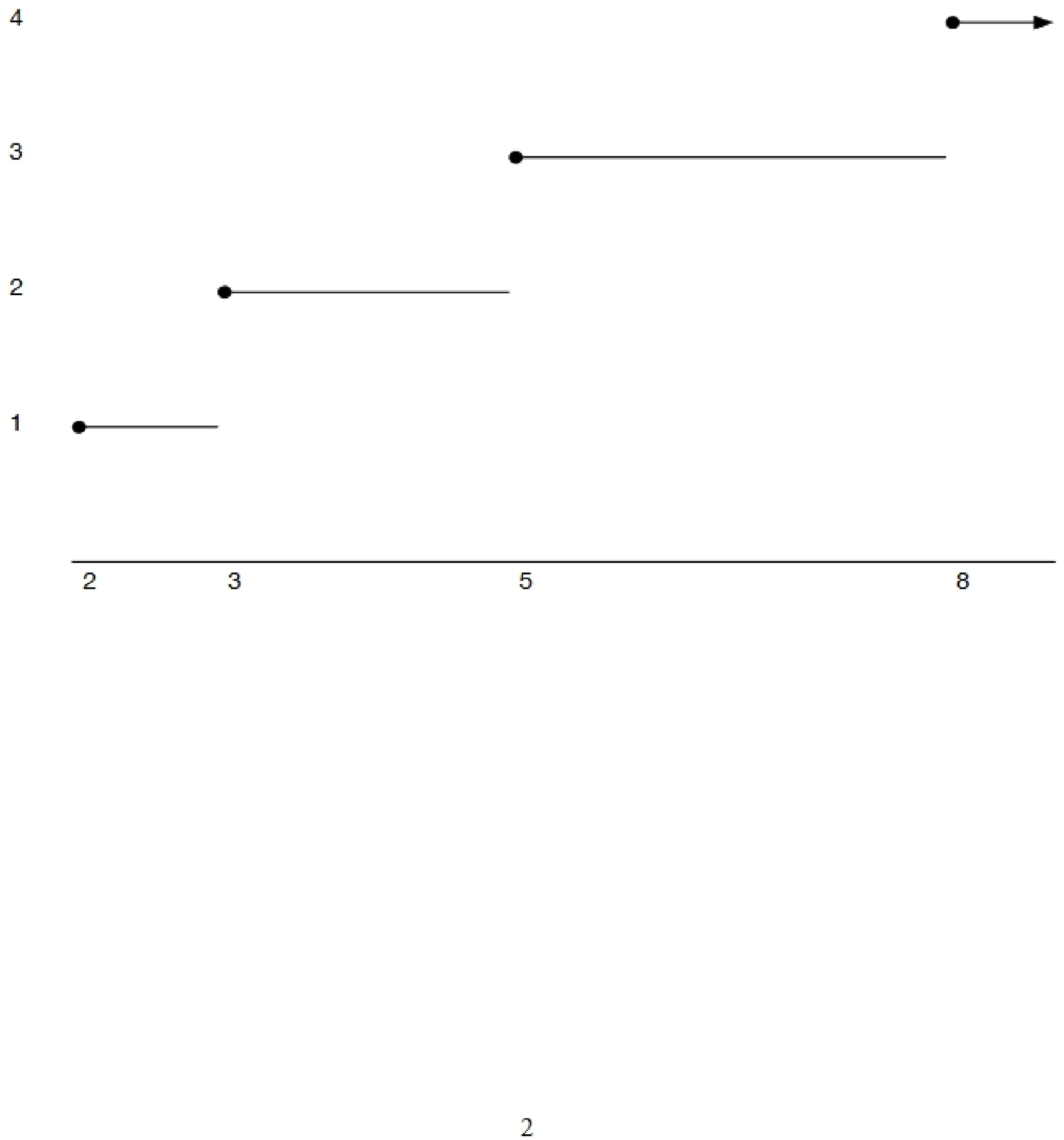
Fibonacci and Euclid: second encounter via Lamé

Theorem **(Lamé)**. *For any $k \geq 1$, if Euclid's algorithm takes k or more trips to compute $\gcd(m, n)$, where $m \geq n$, then $n \geq f_{k+1}$.*

Remark *This shows that the Fibonacci numbers are the worst case for Euclid's algorithm. To see why, look first at the contrapositive version of Lamé.*

Theorem *If $n < f_{k+1}$, then Euclid's algorithm takes at most $k - 1$ trips through the loop.*

The picture of this is:



Proof of Lamé

By strong induction on k.

Basis: $k = 1$. If we went through the loop once then certainly $n \geq 1 = f_2$. And when $k = 2$ we went through the loop twice, so $n > 1$, and thus $n \geq 2 = f_3$.

Induction step: Assume for all integers $\leq k$ that if we go through the loop k times, then $n \geq f_{k+1}$. We must prove the same statement with k replaced by $k + 1$. Suppose that it takes $k + 1$ trips to compute $gcd(m, n)$. Write out the first two trips

$$\begin{aligned} \gcd(m, n) &= \gcd(n, m \bmod n) \\ &= \gcd(m \bmod n, n \bmod (m \bmod n)) \end{aligned}$$