Algorithme d'Euclid

```
E1  Algo gcd(a, b)
E2  tandis que (b ≠ 0)
E3      c ← a mod b
E4      a ← b
E5      b ← c
E6  retourner a
```

(on impose $b \leq a$)

Déf. Nombres Fibonacci : $F(0) = 0; F(1) = 1; F(n+1) = F(n) + F(n-1)$ si $n > 0$.

Lemme. Si $x < F(n+1)$ et $y \geq F(n)$ alors $x \bmod y < F(n-1)$.

Analyse de l'algorithme : dénotons les valeurs des variables (juste après ligne E3) par $\langle a_1, b_1, c_1 \rangle$, $\langle a_2, b_2, c_2 \rangle$, ...

Algorithme d'Euclid 2

Lemme. Si $b_i < F(n)$ alors $b_{i+1} < F(n-1)$ ou $b_{i+2} < F(n-2)$.

Si $b_i < F(3) = 2$, alors le boucle ne sera plus executé.

$\Rightarrow$ Si $b < F(n)$, alors le boucle est executé $\leq (n-2)$ fois.

Notez que $n = O(\log b)$ suffit pour cette borne : $F(n) \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$

$\Rightarrow$ temps de calcul (opérations sur bits) est de

$$O(\log^3 a)$$

donc polynomial dans la taille de l'entrée.

[ici, on suppose que calculer $a \bmod b$ prend $O(\log^2 a)$ temps . . . en fait, on peut le faire plus rapidement]