Annexe F

# *Arithmétique*

Hugo Salou MPI*

Dernière mise à jour le 22 mars 2023

Un des premiers algorithmes codé est l'algorithme d'Euclide pour calculer le PGCD. Pour $a \neq 0$, on a $a \wedge 0 = a$ et $a \wedge b = b \wedge (a \bmod b)$. On peut le coder en OCaml avec la fonction `euclid` suivante.

```
let rec euclid (a: int) (b: int): int =
  (* Hyp: a >= b et a != 0 *)
  if b = 0 then a
  else euclide b (a mod b)
```

Code 1 – Algorithme d'Euclide calculant le PGCD

Quelle est la complexité de cet algorithme ? On représente le nombre d'appels récursifs à `euclid`, et on devine une courbe logarithmique. En notant $(u_n)$ les divisions euclidiennes réalisées et $(q_n)$ les quotients, ainsi, on $u_n = q_{n-1} \cdot u_{n-1} + u_{n-2}$. Alors, $\texttt{euclid}(u_n, u_{n-1}) = \cdots = \texttt{euclid}(u_3, u_2) = \texttt{euclid}(u_2, u_1) = \texttt{euclid}(u_1, u_0)$.

En fixant la complexité, on cherche les valeurs de $(u_n)$ maximisant les appels récursifs. On peut montrer par récurrence que si $\texttt{euclid}(a, b)$ conduit à $n$ appels récursifs de `euclid`, alors $a \geqslant F_n$ et $b \geqslant F_{n-1}$, où $(F_n)_{n\in\mathbb{N}}$ est la suite de Fibonacci.

En effet, soit un tel couple $(a, b)$. Alors, $(b, a \bmod b)$ conduit à $n-1$ appels récursifs donc $b \geqslant F_{n-1}$ et $a \bmod b \geqslant F_{n-2}$ par hypothèse de recurrence. Et, $a = bq + (a \bmod b)$ et donc $a \geqslant F_{n-1} + F_{n-2} = F_n$.

De plus, pour tout $n \in \mathbb{N} \setminus \{0, 1\}$, $F_n \geqslant \varphi^{n-2}$ où $\varphi$ est le nombre d'or.[1] En effet, $F_2 = 1 \geqslant \varphi^0 = 1$ et $F_3 = 2 \geqslant \varphi^1 = \varphi = (1 + \sqrt{5})/2$. Et, $F_n = F_{n-1} + F_{n-2} \geqslant \varphi^{n-3} + \varphi^{n-4} \geqslant \varphi^{n-4}(1 + \varphi) \geqslant \varphi^{n-2}$.

Soient $(p, q)$, où $p \geqslant q$, une entrée de l'algorithme d'Euclide. Si l'appel $\texttt{euclid}(p, q)$ conduit à plus de $\lceil \log_\varphi p \rceil + 4$ appels, alors $p \geqslant F_{\lceil \log_\varphi p \rceil + 4} \geqslant \varphi^{\lceil \log_\varphi p \rceil + 4 - 2} > \varphi^{\log_\varphi p} = p$, ce qui est absurde.

Ceci conduit à une complexité en $\mathcal{O}(\log p)$.

Soit $n$ un entier premier. Pour l'algorithme RSA, on cherche un inverse de $a \in \mathbb{Z}/n\mathbb{Z}$ : on cherche $b \in \mathbb{Z}/n\mathbb{Z}$ tel que $ab \equiv 1\ [1]$. D'après le théorème de Bézout, on a $au + nv = 1$ car $a \wedge n = 1$. L'inverse est $v$. D'où l'importance des coefficients de Bézout.

Comment calculer les coefficients de Bézout ? On peut utiliser l'algorithme d'Euclide. On pose $r_n$ la valeur de a après $n$ appels récursifs.

| $r_i$ | $u_i$ | | $v_i$ | |
|---|---|---|---|---|
| $r_0 = a$ | $1$ | $a$ | $0$ | $b$ |
| $r_1 = b$ | $0$ | $b$ | $1$ | $a$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $r_{i-2}$ | $u_{i-2}$ | $a$ | $v_{i-2}$ | $b$ |
| $r_{i-1}$ | $u_{i-1}$ | $a$ | $v_{i-1}$ | $b$ |

Table 1 – Valeurs de $r_i$ avec invariant $r_i = au_i + bv_i$

Alors,

$$\begin{aligned} r_i &= u_{i-2}a + v_{i-2}b - (r_{i-2}/r_{i-1})(u_{i-1}a + v_{i-1}b) \\ &= \big(u_{i-2} - (r_{i-2}/r_{i-1})u_{i-1}\big)a + \big(v_{i-2} - (r_{i-2}/r_{i-1})v_{i-1}\big)b \end{aligned}$$

Ainsi, on a bien $\mathrm{pgcd}(a, b) = u_{n-1}a + v_{n-1}b$.

1. C'est la solution positive de $X^2 - X - 1 = 0$.