

TD/TP ALGORITHME D'EUCLIDE

**Exercice 1 — Algorithme d'Euclide binaire** — Soient  $u, v \geq 1$  des entiers avec  $v$  impair.

1. Vérifier que l'algorithme suivant renvoie le pgcd de  $u$  et  $v$  :  
Si  $u$  est pair alors faire  $(u, v) \leftarrow (u/2, v)$  ;  
Si  $u$  est impair  $> v$  alors faire  $(u, v) \leftarrow \left( \frac{u-v}{2}, v \right)$  ;  
Si  $u$  est impair  $< v$  alors faire  $(u, v) \leftarrow \left( \frac{v-u}{2}, u \right)$  ;  
Si  $u = v$  alors renvoyer  $u$ .
2. Estimer le nombre d'étapes et le coût de cet algorithme.

**Exercice 2 — Comparaison des algorithmes** —

1. Implémenter en Sage l'algorithme d'Euclide classique.
2. Implémenter en Sage l'algorithme d'Euclide binaire (voir exercice 1).
3. Comparer expérimentalement les nombres d'étapes et les temps d'exécution de ces deux algorithmes.

**Exercice 3 — Étude expérimentale du nombre d'étapes** — Si  $m$  et  $n$  sont deux entiers  $\geq 1$ , on note  $t(m, n)$  le nombre de divisions euclidiennes nécessaires pour calculer le pgcd de  $m$  et  $n$  (on commence par diviser  $m$  par  $n$  et on s'arrête lorsque le reste est nul, ainsi  $t(2, 1) = 1$  et  $t(1, 2) = 2$ ).

1. Écrire une procédure Sage qui calcule  $t(m, n)$ .
2. Si  $(F_n)_{n \geq 0}$  est la suite de Fibonacci (définie par  $F_0 = 0$ ,  $F_1 = 1$  et  $F_{n+2} = F_n + F_{n+1}$  pour tout  $n \geq 0$ ), vérifier  $t(F_{n+2}, F_{n+1}) = n$  pour tout  $1 \leq n \leq 50$ .
3. Écrire une procédure Sage qui vérifie que le maximum de  $t(m, n)$  pour des entiers  $1 \leq m, n \leq N$  est obtenu pour le couple  $(F_r, F_{r+1})$ , où  $r$  est le plus grand entier tel que  $F_{r+1} \leq N$ .
4. On considère les fonctions  $T$  et  $\tau$  définies par

$$T(n) = \frac{1}{n} \sum_{1 \leq k \leq n} t(k, n) \quad \tau(n) = \frac{1}{\varphi(n)} \sum_{\substack{1 \leq k \leq n \\ (k, n) = 1}} t(k, n)$$

où  $\varphi$  est l'indicatrice d'Euler. Tracer les graphes de  $T$  et  $\tau$  sur un même graphique. Que constatez-vous? Vérifier graphiquement qu'il existe un entier  $c$  pour lequel  $0,843 \ln(n) + 1,47$  est une bonne approximation de  $\tau(n)$ .

On considère la variante de l'algorithme d'Euclide dans laquelle le reste d'une division euclidienne  $m = qn + r$  vérifie  $|r| \leq n/2$ . On note  $t'(m, n)$  le nombre de pas nécessaires pour calculer le pgcd de  $m$  et  $n$  avec cette variante.

5. Déterminer  $t'(F_{n+2}, F_{n+1})$ .
6. Reprendre les questions 1, 3 et 4 pour la fonction  $t'$  (on notera  $T'$  et  $\tau'$  les fonctions correspondantes).
7. Tracer sur un même graphique les fonctions  $\tau$  et  $\tau'$ . Que constatez-vous ?

**Exercice 4 — Algorithme d'Euclide étendu —**

Soient  $u, v \geq 1$  des entiers. On note  $d = \text{pgcd}(u, v)$ .

1. On suppose que  $d \neq \min(u, v)$ . Montrer qu'il existe un unique couple d'entiers  $(x, y)$  tels que  $d = ux + vy$  et

$$|x| \leq \left\lfloor \frac{v}{2d} \right\rfloor, \quad |y| \leq \left\lfloor \frac{u}{2d} \right\rfloor,$$

où  $\lfloor \cdot \rfloor$  désigne la partie entière. On dit alors que la relation de Bézout est minimale.

On rappelle l'algorithme d'Euclide étendu appliqué à deux entiers  $u, v \geq 1$ , qui renvoie un triplet  $(d, x, y)$  avec  $d = \text{pgcd}(u, v)$  et  $d = ux + vy$ .

$(r_0, r_1) \leftarrow (u, v)$  ;

$(x_0, x_1) \leftarrow (1, 0)$  ;

$k \leftarrow 0$  ;

Tant que  $r_{k+1} \neq 0$  faire :

Écrire  $r_k = q_k r_{k+1} + r_{k+2}$  avec  $0 \leq r_{k+2} < r_{k+1}$  ;

$x_{k+2} \leftarrow x_k - q_k x_{k+1}$  ;

$k \leftarrow k + 1$  ;

Renvoyer  $\left( r_k, x_k, \frac{r_k - x_k u}{v} \right)$ .

On notera  $n$  la valeur de  $k$  à la fin de l'algorithme (autrement dit, vérifiant  $r_{n+1} = 0$ ).

2. Montrer  $(-1)^k x_k \geq 0$  pour tout  $0 \leq k \leq n + 1$ .
3. Montrer que la suite  $(|x_k|)_{1 \leq k \leq n+1}$  est croissante.
4. Montrer que  $D_k = (-1)^k (r_k x_{k+1} - r_{k+1} x_k)$  est indépendant de  $k$ .
5. Montrer que  $|x_n| \leq |x_{n+1}|/2$ .
6. Montrer que l'algorithme ci-dessus renvoie la relation de Bézout minimale.

**Exercice 5 — Inverse modulaire —** Soient  $a, b \geq 1$  des entiers premiers entre eux. Estimer le coût du calcul de l'inverse de  $a$  dans  $\mathbb{Z}/b\mathbb{Z}$ .