

---

 ARITHMÉTIQUE
 

---

## 1. ALGORITHME D'EUCLIDE

**Exercice 1.** Implémenter l'algorithme d'Euclide pour les entiers positifs. Illustrer le temps d'exécution de votre fonction. Comparer avec `gcd`. On rappelle que `//` et `%` permettent de calculer le quotient et le reste d'une division euclidienne et que `timeit` permet de mesurer des temps d'exécution.

**Exercice 2.**

- (1) Écrire une fonction qui à l'entrée  $n \geq 1$  associe le  $n$ -ième nombre de Fibonacci  $F_n$ . On rappelle que la suite  $(F_n)$  est définie par :

$$F_1 = F_2 = 1 \text{ et } F_{n+2} = F_{n+1} + F_n \text{ pour } n \geq 1.$$

Vérifier que votre procédure donne un résultat rapide pour des valeurs de  $n$  «raisonnablement grandes» (disons dans le cas où  $n$  a 3 chiffres).

- (2) Vérifier à l'aide de `sage` que  $F_{n+1}$  et  $F_n$  sont premiers entre eux pour tout  $1 \leq n \leq 100$ .
- (3) Modifier votre implémentation de l'algorithme d'Euclide pour qu'apparaisse en sortie de nombre de divisions euclidiennes nécessaires au calcul du pgcd. Vérifier que le calcul de `pgcd(Fn+1, Fn)` nécessite  $n$  divisions euclidiennes, pour  $2 \leq n \leq 100$ .
- (4) Proposer une illustration du coût en divisions euclidiennes de l'algorithme d'Euclide.

**Exercice 3.**

- (1) Écrire l'algorithme d'Euclide étendu pour les entiers positifs. La commande `xgcd` permet de calculer une relation de Bézout avec `sage`, vous pourrez l'utiliser dans la suite du TP.
- (2) Écrire une procédure `inverse_mod(a,m)` permettant de calculer l'inverse d'un entier  $a$  premier avec un entier  $m \geq 2$  donné. Si  $a$  n'est pas premier avec  $m$ , on utilisera la commande

```
raise ValueError("Le nombre {} n'est pas inversible modulo {}".format(a,m))
```

où la méthode `format` appliquée à une chaîne de caractères remplace les `{}` par la valeur des expressions données comme arguments (ici  $a$  et  $m$ ). Comparer la réponse à la commande `inverse_mod(2,4)` avec la réponse habituelle de `sage` en cas d'erreur.

**Exercice 4.** On considère l'algorithme récursif décrit ci-dessous.

Entrées : deux entiers strictement positifs  $a$  et  $b$ .

Sortie :  $R(a, b)$ .

1. si  $a = b$ , on renvoie  $a$ .
2. si  $a$  et  $b$  sont pairs, on renvoie  $2R(a/2, b/2)$ .
3. si  $a$  n'ont pas la même parité, on renvoie  $R(a/2, b)$  si  $a$  est pair et  $R(a, b/2)$  si  $b$  est pair.
5. si  $a$  et  $b$  sont impairs, on renvoie  $R(a, (b - a)/2)$  si  $b > a$  et  $R((a - b)/2, b)$  sinon.

Que calcule cet algorithme? Justifier.

**Exercice 5.** On note  $\phi$  le nombre d'or. Simplifier la fraction

$$\frac{\phi^4 - \phi + 1}{\phi^7 - 1}$$

sans utiliser de formule explicite pour  $\phi$ .

2. ANNEAUX  $\mathbb{Z}/n\mathbb{Z}$ 

**Exercice 6** (Petit théorème de Fermat).

- (1) Écrire une fonction `fermat(p)` qui vérifie le *petit théorème de Fermat* : « si  $p$  est premier alors  $a^p \equiv a \pmod{p}$  pour tout entier  $a$  » pour un nombre premier  $p$  donné.
- (2) Vérifier le théorème pour les premiers  $p \leq 11$ .
- (3) Le nombre 561 est-il premier ? A-t-on  $a^{561} \equiv a \pmod{561}$  pour tout entier  $a$  ? Que peut-on en conclure ?

**Exercice 7.** Le groupe  $(\mathbb{Z}/n\mathbb{Z})^\times$  est-il toujours cyclique ? On sait que le groupe  $(\mathbb{Z}/p\mathbb{Z})^\times$  est cyclique pour  $p$  premier. Illustrer cette propriété avec `sage` et développer (par exemple, en déterminant tous les générateurs et en expliquant comment ils se déduisent les uns des autres).

## 3. RESTES CHINOIS

**Exercice 8** (Restes chinois). Écrire une fonction `systeme_chinois(nu, m)` qui, à partir de listes d'entiers `nu` et `m`, renvoie une solution  $x$  du système de congruences

$$x \equiv \nu_i \pmod{m_i}, \quad 1 \leq i \leq r.$$

en supposant que les  $m_i$  sont 2 à 2 premiers entre eux.

**Exercice 9.** Soit  $P = X^6 + 1$  et  $Q = X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$ .

- (1) Donner deux polynômes de  $\mathbb{Q}[X]$  de degrés les plus petits possibles qui vérifient simultanément :  $R \equiv X^2 [P]$  et  $R \equiv X + 1 [Q]$ .
- (2) Décrire explicitement l'inverse du morphisme

$$\mathbb{Q}[X]/(PQ) \rightarrow \mathbb{Q}[X]/(P) \times \mathbb{Q}[X]/(Q).$$

**Exercice 10.** Déterminer un polynôme  $P \in \mathbb{Q}[X]$  de degré au plus 3 tel que  $P(0) = 1$ ,  $P'(0) = 1$ ,  $P(1) = 1$ ,  $P'(1) = -1$ ,  $P(-1) = 1$  et  $P'(-1) = 0$ .

**Exercice 11.** Un élément  $e$  d'un anneau s'appelle un idempotent si  $e^2 = e$ . Un *système d'idempotents orthogonaux* est une famille d'idempotents  $(e_1, \dots, e_n)$  tels que  $e_i e_j = \delta_{i,j} e_i$  pour tout  $1 \leq i, j \leq n$ . Il est dit *complet* si  $e_1 + \dots + e_n = 1$ .

- (1) Si  $A$  est un anneau isomorphe à un produit d'anneaux  $A_1 \times \dots \times A_n$ , expliquer comment trouver un système d'idempotents orthogonaux complet de  $A$ .
- (2) Écrire un programme qui étant donné un nombre premier  $p$  et  $n$  éléments  $(x_1, \dots, x_n)$  de  $\mathbb{F}_p$  renvoie un système d'idempotent orthogonaux complet  $e_1, \dots, e_n$  ( $e_i e_j = \delta_{i,j} e_i$  pour tout  $i, j$ ) de

$$\frac{\mathbb{F}_p[X]}{\prod_{i=1}^n (X - x_i)}$$

- (3) Trouver un système d'idempotents orthogonaux complet  $\{e_1, e_2, e_3\}$  pour l'anneau  $\mathbb{Z}/30\mathbb{Z}$ .