Chapitre 1

La division euclidienne et ses conséquences

1.1 La division euclidienne

Division euclidienne pour les entiers positifs

Théorème 1.1.1. *Pour $a \in \mathbb{N}$, $b \in \mathbb{N}^*$, il existe un unique couple d'entiers (q, r) avec $a = bq + r$ et $0 \leq r < b$.*

Méthodes algorithmiques :

Méthode naïve :

Entrée : $a \geq 0$ et $b > 0$;
$n \leftarrow 0$; $r \leftarrow a$;
While $(r > b)$ **do**
    $r \leftarrow r - b$;
    $q \leftarrow q + 1$;
done
Retourner q et r ;

Algorithm 1: Division euclidienne, méthode naïve

En dialecte machine (python) :

```
def div_naive(a,b):
  q,r=0,a
```

```
while r>=b:
  q,r=q+1,r-b
return(q,r)
```

Le nombre de boucle de cet algorithme est q ; à chaque étape il y a une soustraction et une comparaison ; pour b fixé la complexité croît linéairement avec a. Si on mesure la taille de l'entier a par le nombre de chiffres de son développement en binaire : $t = \log_2(a)$, alors la *complexité* croît exponentiellement avec la taille de a. On peut faire apparaître expérimentalement cette complexité (test1.py dans le dossier python) et constater la meilleure efficacité de l'algorithme dichotomique (voir le code python de la fonction div_dicho ainsi que test2.py, test3.py).

Division euclidienne pour les entiers relatifs

Théorème 1.1.2. *Pour $a \in \mathbb{Z}$, $b \in \mathbb{Z}^*$, il existe un unique couple d'entiers (q, r) avec $a = bq + r$ et $0 \leq r < |b|$.*

Exercice 1.1.3. Modifier l'algorithme naïf précédent pour effectuer la division euclidienne des entiers relatifs.

1.2 Sous-groupes de $\mathbb{Z}$

Définition 1.2.1. On appelle sous-groupe de $\mathbb{Z}$ tout sous-ensemble H qui contient 0 et qui est stable par addition et soustraction.

Exemples : le sous-groupe engendré par n : $n\mathbb{Z}$; $a\mathbb{Z} + b\mathbb{Z}$; l'intersection de deux sous-groupes.

Théorème 1.2.2. *Tout sous-groupe de $\mathbb{Z}$ est de la forme $n\mathbb{Z}$.*

On dit que n est générateur du sous-groupe.

Proposition 1.2.3. *Tout sous-groupe de $\mathbb{Z}$ a un unique générateur positif.*

1.3 Diviseurs

Proposition 1.3.1. *Pour $a \in \mathbb{Z}$ et $b \in \mathbb{Z}$, il y a équivalence entre b divise a et*

$$a\mathbb{Z} \subset b\mathbb{Z} \ .$$

Proposition 1.3.2. *a) Le générateur positif d de $a\mathbb{Z} + b\mathbb{Z}$ est le PGCD de a et b : c'est le plus grand diviseur commun à a et b.*
b) Les diviseurs communs à a et b sont les diviseurs du PGCD.

Proposition 1.3.3. *a) Le générateur positif de $a\mathbb{Z} \cap b\mathbb{Z}$ est le PPCM de a et b : c'est le plus petit multiple commun positif à a et b.*
b) Les multiples communs à a et b sont les multiples du PPCM.

1.4 L'algorithme d'Euclide

Proposition 1.4.1. *Soient : $a \in \mathbb{Z}$, $b \in \mathbb{Z}$. On a pour tout $m \in \mathbb{Z}$:*

$$PGCD(a, b) = PGCD(b, a - mb) \ .$$

On peut en particulier appliquer la proposition précédente au cas où $m = q$ est le quotient dans la division euclidienne de a par b.

```
Fonction PGCD(a, b) ;
a ← |a| ; b ← |b| ;
If (b > a) then
  | a ↔ b ;
end If
r ← a mod b (reste de la division euclidienne) ;
If (r est nul) then
  | Retourner b ;
else
  | Retourner PGCD(b, r) ;
end If
```

Algorithm 2: Algorithme d'Euclide, forme récursive

Fin du cours du 23/01

```
Fonction PGCD(a, b) ;
a ← |a| ; b ← |b| ;
If (b > a) then
  | a ↔ b ;
end If
While (b est non nul) do
    | r ← a mod b (reste de la division euclidienne) ;
    | a ← b ;b ← r ;
done
Retourner a ;
```

Algorithm 3: Algorithme d'Euclide, forme non récursive

1.5 Théorème de Bezout

Rappel : Deux entiers relatifs sont premiers entre eux si et seulement si leur PGCD est égal à 1. Un entier naturel p est premier si et seulement s'il a exactement deux diviseurs positifs : 1 et p.

Théorème 1.5.1 (Bezout)**.** *Deux entiers a et b sont premiers entre eux si et seulement s'il existe un couple (u, v) tel que $au + bv = 1$.*

Corollaire 1.5.2 (théorème de Gauss)**.** *Soient a, b et c trois entiers relatifs. Si a divise le produit bc et est premier avec b, alors a divise c.*

Corollaire 1.5.3 (théorème d'Euclide)**.** *Si un entier premier divise un produit, alors il divise un des facteurs.*

Equation $ax + by = c$**.**

Proposition 1.5.4. *a) L'équation $ax + by = c$ a des solutions si et seulement si le PGCD de a et b divise c.*
b) Lorsque a et b sont premiers entre eux, si (x_0, y_0) est une solution, alors les solutions sont : $(x = x_0 - kb, y = y_0 + ka),\ k \in \mathbb{Z}$.

1.6 Algorithme d'Euclide étendu aux coefficients de Bezout

[En entrée : $a > b \geq 0$; la fonction retourne 3 entiers]
Fonction PGCDE(a, b) ;
If (b est nul) **then**
  Retourner $(a, 1, 0)$;
else
  Effectuer la division euclidienne de a par b ; $r \leftarrow$ reste ;$q \leftarrow$ quotient ;
  $(d, u', v') \leftarrow PGCDE(b, r)$;
  $u \leftarrow v'$; $v \leftarrow (u' - qv')$;
  Retourner (d, u, v) ;
end If

Algorithm 4: Algorithme d'Euclide, forme récursive

En dialecte machine (python) :

```
def pgcder(a,b): # PGCD étendu récursif avec entrées positives
    permute=(b>a)
    if (permute):  # mettre en ordre croissant
        a,b=b,a
    if (b==0):    # sortie de récursion
        return a,1,0
    else:
        q,r=a/b,a%b  # division euclidienne
        d,u1,v1=pgcder(b,r) # appel récursif
        u,v=v1,u1-q*v1
        if permute:
            u,v=v,u
        return d,u,v
```

Exercice 1.6.1. Ecrire un algorithme d'Euclide étendu non récursif.

Fin du cours du 30/01

1.7 Décomposition en facteurs premiers

Théorème 1.7.1. *Tout entier naturel supérieur ou égal à* 2 *s'écrit de manière unique comme un produit de facteurs premiers.*

Proposition 1.7.2. *Etant donnés deux entiers a et b supérieur à* 1*, leur PGCD est le produit des facteurs premiers communs avec le plus faible exposant, et leur PPCM est le produit de tous les facteurs premiers avec le plus fort exposant.*

Corollaire 1.7.3. *Etant donnés deux entiers a et b supérieur à* 1*, on a :*

$$PGCD(a,b) \times PPCM(a,b) = ab \ .$$

Exercice 1.7.4. Ecrire un algorithme pour tester si un nombre est premier.

Exercice 1.7.5. Ecrire un algorithme pour décomposer en facteurs premiers. On pourra utiliser un tableau T contenant les nombres premiers successifs $T[1] = 2, \ldots, T[N] = P$. On traitera en entrée les entiers inférieurs ou égaux à P^2 ; en sortie on obtient les facteurs premiers avec leur exposant.