Introduction à la Cryptologie

Chapitre 3 : Euclide–Bézout et applications

Michael Eisermann (Institut Fourier, UJF Grenoble)

Année 2008-2009
IF / IMAG, Master 1, S1-S2

document mis à jour le 7 juillet 2009

www-fourier.ujf-grenoble.fr/~eiserm/cours # crypto

Objectifs de ce chapitre

Développement mathématique :

- Préciser le vocabulaire : divisibilité, nombres premiers, pgcd.
- Établir les lemmes de Gauss et d'Euclide, puis illustrer leur utilité.
- Établir la décomposition en facteurs premiers : existence et unicité.

Objectifs de ce chapitre

Développement mathématique :

- Préciser le vocabulaire : divisibilité, nombres premiers, pgcd.
- Établir les lemmes de Gauss et d'Euclide, puis illustrer leur utilité.
- Établir la décomposition en facteurs premiers : existence et unicité.

Développement algorithmique :

- Établir l'algorithme d'Euclide : correction et complexité.
- Établir l'algorithme d'Euclide–Bézout : correction et complexité.
- Discuter les problèmes liés à la factorisation de grands entiers.

Sommaire

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\operatorname{pgcd}(a,b) = \prod_{p \text{ premier}} p^{\min(\nu_p(a),\nu_p(b))},$$
$$\operatorname{ppcm}(a,b) = \prod_{p \text{ premier}} p^{\max(\nu_p(a),\nu_p(b))}.$$

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\operatorname{pgcd}(a,b) = \prod_{p \text{ premier}} p^{\min(\nu_p(a),\nu_p(b))},$$

$$\operatorname{ppcm}(a,b) = \prod_{p \text{ premier}} p^{\max(\nu_p(a),\nu_p(b))}.$$

Bien que importantes pour la théorie, elles sont inefficace pour le calcul du pgcd ou du ppcm :

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\begin{aligned} \operatorname{pgcd}(a,b) &= \prod_{p \text{ premier}} p^{\min(\nu_p(a),\nu_p(b))}, \\ \operatorname{ppcm}(a,b) &= \prod_{p \text{ premier}} p^{\max(\nu_p(a),\nu_p(b))}. \end{aligned}$$

Bien que importantes pour la théorie, elles sont inefficace pour le calcul du pgcd ou du ppcm : il faudrait d'abord factoriser a et b.

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\begin{aligned} \operatorname{pgcd}(a,b) &= \prod_{p \text{ premier}} p^{\min(\nu_p(a),\nu_p(b))}, \\ \operatorname{ppcm}(a,b) &= \prod_{p \text{ premier}} p^{\max(\nu_p(a),\nu_p(b))}. \end{aligned}$$

Bien que importantes pour la théorie, elles sont inefficace pour le calcul du pgcd ou du ppcm : il faudrait d'abord factoriser a et b. Or, pour les grands entiers, la factorisation est un problème très dur.

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\begin{aligned} \operatorname{pgcd}(a,b) &= \prod_{p \text{ premier}} p^{\min(\nu_p(a),\nu_p(b))}, \\ \operatorname{ppcm}(a,b) &= \prod_{p \text{ premier}} p^{\max(\nu_p(a),\nu_p(b))}. \end{aligned}$$

Bien que importantes pour la théorie, elles sont inefficace pour le calcul du pgcd ou du ppcm : il faudrait d'abord factoriser a et b. Or, pour les grands entiers, la factorisation est un problème très dur.

Heureusement il existe une méthode très efficace, l'algorithme d'Euclide, qui évite entièrement le problème de factorisation.

Motivation

Le théorème fondamental de l'arithmétique dit que tout entier positif a s'écrit de manière unique comme produit de nombres premiers positifs :

$$a = 2^{\nu_2} \cdot 3^{\nu_3} \cdot 5^{\nu_5} \cdot 7^{\nu_7} \cdots = \prod_{p \text{ premier}} p^{\nu_p}$$

avec des exposants $\nu_p = \nu_p(a) \in \mathbb{N}$ dont tous sauf un nombre fini sont nuls.

Ce théorème est l'objectif mathématique de ce chapitre.

Pour le pgcd et le ppcm on obtient alors les formules

$$\begin{aligned} \operatorname{pgcd}(a,b) &= \prod_{p \text{ premier}} p^{\min(\nu_p(a), \nu_p(b))}, \\ \operatorname{ppcm}(a,b) &= \prod_{p \text{ premier}} p^{\max(\nu_p(a), \nu_p(b))}. \end{aligned}$$

Bien que importantes pour la théorie, elles sont inefficace pour le calcul du pgcd ou du ppcm : il faudrait d'abord factoriser a et b. Or, pour les grands entiers, la factorisation est un problème très dur.

Heureusement il existe une méthode très efficace, l'algorithme d'Euclide, qui évite entièrement le problème de factorisation.

Cet algorithme est l'objectif algorithmique de ce chapitre.

Sommaire

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre :

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)
- $a \mid b$ et $b \mid a$ impliquent $a = \pm b$ (antisymétrie)

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)
- $a \mid b$ et $b \mid a$ impliquent $a = \pm b$ (antisymétrie)
- $a \mid b$ et $b \mid c$ impliquent $a \mid c$ (transitivité)

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)
- $a \mid b$ et $b \mid a$ impliquent $a = \pm b$ (antisymétrie)
- $a \mid b$ et $b \mid c$ impliquent $a \mid c$ (transitivité)

De plus, pour tout $a, b, c \in \mathbb{Z}$ on a :

- $1 \mid a$ et $a \mid 0$.

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)
- $a \mid b$ et $b \mid a$ impliquent $a = \pm b$ (antisymétrie)
- $a \mid b$ et $b \mid c$ impliquent $a \mid c$ (transitivité)

De plus, pour tout $a, b, c \in \mathbb{Z}$ on a :

- $1 \mid a$ et $a \mid 0$.
- Si $a \mid b$, alors $a \mid bc$.

Divisibilité

On note $a\mathbb{Z} = \{ac : c \in \mathbb{Z}\}$ l'ensemble des multiples de a.

Définition (divisibilité)

Étant donnés $a, b \in \mathbb{Z}$ on dit que a *divise* b, noté $a \mid b$, s'il existe $c \in \mathbb{Z}$ tel que $ac = b$. Autrement dit, on a $a \mid b$ si et seulement si $b \in a\mathbb{Z}$.

Remarque

La divisibilité définit un pré-ordre : pour tout $a, b, c \in \mathbb{Z}$ on a :

- $a \mid a$ (réflexivité)
- $a \mid b$ et $b \mid a$ impliquent $a = \pm b$ (antisymétrie)
- $a \mid b$ et $b \mid c$ impliquent $a \mid c$ (transitivité)

De plus, pour tout $a, b, c \in \mathbb{Z}$ on a :

- $1 \mid a$ et $a \mid 0$.
- Si $a \mid b$, alors $a \mid bc$.
- Si $a \mid b$ et $a \mid c$, alors $a \mid b + c$.

Définition

Un entier $p \in \mathbb{Z}$ est dit *nombre premier* si $p \notin \{+1, -1\}$
et si les seuls diviseurs de p sont ± 1 et $\pm p$.

Nombres premiers

Définition

Un entier $p \in \mathbb{Z}$ est dit *nombre premier* si $p \notin \{+1, -1\}$
et si les seuls diviseurs de p sont ± 1 et $\pm p$.

Remarque

Si p est premier, alors $-p$ l'est aussi. Par *nombre premier* on sous-entendra toujours *nombre premier positif*.

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a,

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1) \cap \cdots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \dots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \dots, a_n \in \mathbb{Z}$.

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \ldots, a_n) = \mathscr{D}(a_1) \cap \cdots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \ldots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \ldots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \ldots, a_n \in \mathbb{Z}$ si

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \ldots, a_n) = \mathscr{D}(a_1) \cap \cdots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \ldots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \ldots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \ldots, a_n \in \mathbb{Z}$ si

- d est un diviseur commun, c'est-à-dire que $d \mid a_1, \ldots, d \mid a_n$, et

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1) \cap \dots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \dots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \dots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \dots, a_n \in \mathbb{Z}$ si

- d est un diviseur commun, c'est-à-dire que $d \mid a_1, \dots, d \mid a_n$, et
- d en est un plus grand : si $c \mid a_1, \dots, c \mid a_n$ alors $c \mid d$.

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1) \cap \cdots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \dots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \dots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \dots, a_n \in \mathbb{Z}$ si

- d est un diviseur commun, c'est-à-dire que $d \mid a_1, \dots, d \mid a_n$, et
- d en est un plus grand : si $c \mid a_1, \dots, c \mid a_n$ alors $c \mid d$.

⚠ On utilise le pré-ordre $\mid$ induit par la divisibilité (et non l'ordre $\leq$).

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1) \cap \dots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \dots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \dots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \dots, a_n \in \mathbb{Z}$ si

- d est un diviseur commun, c'est-à-dire que $d \mid a_1, \dots, d \mid a_n$, et
- d en est un plus grand : si $c \mid a_1, \dots, c \mid a_n$ alors $c \mid d$.

⚠ On utilise le pré-ordre $\mid$ induit par la divisibilité (et non l'ordre $\leq$).

Remarque

Si d est un pgcd de $a_1, \dots, a_n$, alors son opposé $-d$ en est un aussi.
La non-unicité nous oblige de dire « un pgcd » au lieu de « le pgcd ».
Dans $\mathbb{Z}$ on sous-entendra par *le pgcd* toujours *le pgcd positif* (ou nul).

Définition du pgcd

On note $\mathscr{D}(a) = \{d \in \mathbb{Z} : d \mid a\}$ l'ensemble des diviseurs de a, puis

$$\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1) \cap \dots \cap \mathscr{D}(a_n) = \{d \in \mathbb{Z} : d \mid a_1, \dots, d \mid a_n\}$$

l'ensemble des diviseurs communs de $a_1, \dots, a_n \in \mathbb{Z}$.

Définition

On dit que $d \in \mathbb{Z}$ est un *pgcd* de $a_1, \dots, a_n \in \mathbb{Z}$ si

- d est un diviseur commun, c'est-à-dire que $d \mid a_1, \dots, d \mid a_n$, et
- d en est un plus grand : si $c \mid a_1, \dots, c \mid a_n$ alors $c \mid d$.

⚠ On utilise le pré-ordre $\mid$ induit par la divisibilité (et non l'ordre $\leq$).

Remarque

Si d est un pgcd de $a_1, \dots, a_n$, alors son opposé $-d$ en est un aussi.
La non-unicité nous oblige de dire « un pgcd » au lieu de « le pgcd ».
Dans $\mathbb{Z}$ on sous-entendra par *le pgcd* toujours *le pgcd positif* (ou nul).

Cette convention assure l'unicité du pgcd et servira de spécification dans nos algorithmes. Il reste encore à montrer l'existence du pgcd.

Rappel

Proposition

L'ensemble $\mathbb{N}$ *muni de sa relation d'ordre* $\leq$ *est bien ordonné :*
Tout sous-ensemble non vide $X \subset \mathbb{N}$ *admet un plus petit élément, c'est-à-dire qu'il existe* $m \in X$ *tel que* $m \leq x$ *pour tout* $x \in X$*.*

Proposition

L'ensemble $\mathbb{N}$ muni de sa relation d'ordre $\leq$ est bien ordonné :
Tout sous-ensemble non vide $X \subset \mathbb{N}$ admet un plus petit élément, c'est-à-dire qu'il existe $m \in X$ tel que $m \leq x$ pour tout $x \in X$.

Corollaire

Dans $\mathbb{N}$ il n'existe pas de suite infinie décroissante $x_0 > x_1 > x_2 > \dots$.

Autrement dit, dans $\mathbb{N}$ toute suite décroissante s'arrête.

Proposition

L'ensemble $\mathbb{N}$ muni de sa relation d'ordre $\leq$ est bien ordonné :
Tout sous-ensemble non vide $X \subset \mathbb{N}$ admet un plus petit élément, c'est-à-dire qu'il existe $m \in X$ tel que $m \leq x$ pour tout $x \in X$.

Corollaire

Dans $\mathbb{N}$ il n'existe pas de suite infinie décroissante $x_0 > x_1 > x_2 > \dots$.

Autrement dit, dans $\mathbb{N}$ toute suite décroissante s'arrête.

Démonstration. Supposons par absurde que $f\colon \mathbb{N} \to \mathbb{N}$, $n \mapsto x_n$, était décroissante, c'est-à-dire que $x_n > x_{n+1}$ pour tout $n \in \mathbb{N}$.

Rappel

Proposition

L'ensemble $\mathbb{N}$ muni de sa relation d'ordre $\leq$ est bien ordonné :
Tout sous-ensemble non vide $X \subset \mathbb{N}$ admet un plus petit élément, c'est-à-dire qu'il existe $m \in X$ tel que $m \leq x$ pour tout $x \in X$.

Corollaire

Dans $\mathbb{N}$ il n'existe pas de suite infinie décroissante $x_0 > x_1 > x_2 > \dots$.

Autrement dit, dans $\mathbb{N}$ toute suite décroissante s'arrête.

Démonstration. Supposons par absurde que $f\colon \mathbb{N} \to \mathbb{N}$, $n \mapsto x_n$, était décroissante, c'est-à-dire que $x_n > x_{n+1}$ pour tout $n \in \mathbb{N}$.

L'ensemble $X = \{x_n \mid n \in \mathbb{N}\}$ admet un plus petit élément, x_m pour un certain $m \in \mathbb{N}$. Ceci contredit l'hypothèse que $x_m > x_{m+1}$. $\square$

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers.

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \quad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0$$

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$
\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}
$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \qquad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0
$$

Proposition

Le dernier reste non nul, noté ici r_n, est un pgcd de a et b.

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \qquad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0$$

Proposition

Le dernier reste non nul, noté ici r_n, est un pgcd de a et b.

Démonstration. Par construction on a $r_{n+1} = 0$, donc r_n divise r_{n-1}.

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \quad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0$$

Proposition

Le dernier reste non nul, noté ici r_n, est un pgcd de a et b.

Démonstration. Par construction on a $r_{n+1} = 0$, donc r_n divise r_{n-1}.
En remontant, on voit que r_n divise aussi $r_{n-2}, \ldots, r_2, r_1 = b, r_0 = a$.

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \qquad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0$$

Proposition

Le dernier reste non nul, noté ici r_n, est un pgcd de a et b.

Démonstration. Par construction on a $r_{n+1} = 0$, donc r_n divise r_{n-1}.
En remontant, on voit que r_n divise aussi $r_{n-2}, \ldots, r_2, r_1 = b, r_0 = a$.
Ceci montre que r_n est un diviseur commun de a et b.

Existence du pgcd : première version de l'algorithme

Soit $a, b \in \mathbb{Z}$ deux entiers. On itère la division euclidienne :

$$
\begin{array}{llll}
r_0 \leftarrow a & & & \\
r_1 \leftarrow b & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\quad \ldots & & \ldots & \ldots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}|
\end{array}
$$

Tant que $r_n \neq 0$ on itère jusqu'à un reste nul :

$$
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n \qquad \text{càd} \qquad r_{n-1} = r_n q_n + r_{n+1}, \qquad r_{n+1} = 0
$$

Proposition

Le dernier reste non nul, noté ici r_n, est un pgcd de a et b.

Démonstration. Par construction on a $r_{n+1} = 0$, donc r_n divise r_{n-1}.
En remontant, on voit que r_n divise aussi $r_{n-2}, \ldots, r_2, r_1 = b, r_0 = a$.
Ceci montre que r_n est un diviseur commun de a et b.
Si un entier c divise a et b, alors c divise $r_0, r_1, r_2, \ldots, r_n$. $\square$

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \ldots, a_n) = \mathscr{D}(a_1, \mathrm{pgcd}(a_2, \ldots, a_n))$.

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1, \mathrm{pgcd}(a_2, \dots, a_n))$.
Ainsi $\mathrm{pgcd}(a_1, \dots, a_n) = \mathrm{pgcd}(a_1, \mathrm{pgcd}(a_2, \dots, a_n))$. $\square$

Observation

On a $\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1, \mathrm{pgcd}(a_2, \dots, a_n))$.
Ainsi $\mathrm{pgcd}(a_1, \dots, a_n) = \mathrm{pgcd}(a_1, \mathrm{pgcd}(a_2, \dots, a_n))$. $\square$

Il suffit donc de savoir calculer le pgcd de deux entiers.

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$.
Ainsi $\operatorname{pgcd}(a_1, \dots, a_n) = \operatorname{pgcd}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$. □

Il suffit donc de savoir calculer le pgcd de deux entiers.

Observation

Pour tout $a, b, c \in \mathbb{Z}$ on a $\mathscr{D}(a, b) = \mathscr{D}(b, a) = \mathscr{D}(b, a - bc)$.

Observation

On a $\mathscr{D}(a_1, \ldots, a_n) = \mathscr{D}(a_1, \operatorname{pgcd}(a_2, \ldots, a_n))$.
Ainsi $\operatorname{pgcd}(a_1, \ldots, a_n) = \operatorname{pgcd}(a_1, \operatorname{pgcd}(a_2, \ldots, a_n))$. $\square$

Il suffit donc de savoir calculer le pgcd de deux entiers.

Observation

Pour tout $a, b, c \in \mathbb{Z}$ on a $\mathscr{D}(a, b) = \mathscr{D}(b, a) = \mathscr{D}(b, a - bc)$.
Ainsi $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a) = \operatorname{pgcd}(b, a - bc)$. $\square$

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \ldots, a_n) = \mathscr{D}(a_1, \operatorname{pgcd}(a_2, \ldots, a_n))$.
Ainsi $\operatorname{pgcd}(a_1, \ldots, a_n) = \operatorname{pgcd}(a_1, \operatorname{pgcd}(a_2, \ldots, a_n))$. □

Il suffit donc de savoir calculer le pgcd de deux entiers.

Observation

Pour tout $a, b, c \in \mathbb{Z}$ on a $\mathscr{D}(a, b) = \mathscr{D}(b, a) = \mathscr{D}(b, a - bc)$.
Ainsi $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a) = \operatorname{pgcd}(b, a - bc)$. □

C'est l'observation clé pour l'algorithme d'Euclide.

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$.
Ainsi $\operatorname{pgcd}(a_1, \dots, a_n) = \operatorname{pgcd}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$. $\square$

Il suffit donc de savoir calculer le pgcd de deux entiers.

Observation

Pour tout $a, b, c \in \mathbb{Z}$ on a $\mathscr{D}(a, b) = \mathscr{D}(b, a) = \mathscr{D}(b, a - bc)$.
Ainsi $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a) = \operatorname{pgcd}(b, a - bc)$. $\square$

C'est l'observation clé pour l'algorithme d'Euclide.

Observation

On a finalement le cas trivial $\operatorname{pgcd}(a, 0) = \operatorname{pgcd}(a) = |a|$. $\square$

Propriétés du pgcd

Observation

On a $\mathscr{D}(a_1, \dots, a_n) = \mathscr{D}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$.
Ainsi $\operatorname{pgcd}(a_1, \dots, a_n) = \operatorname{pgcd}(a_1, \operatorname{pgcd}(a_2, \dots, a_n))$. $\square$

Il suffit donc de savoir calculer le pgcd de deux entiers.

Observation

Pour tout $a, b, c \in \mathbb{Z}$ on a $\mathscr{D}(a, b) = \mathscr{D}(b, a) = \mathscr{D}(b, a - bc)$.
Ainsi $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a) = \operatorname{pgcd}(b, a - bc)$. $\square$

C'est l'observation clé pour l'algorithme d'Euclide.

Observation

On a finalement le cas trivial $\operatorname{pgcd}(a, 0) = \operatorname{pgcd}(a) = |a|$. $\square$

C'est la condition d'arrêt dans l'algorithme d'Euclide.

Algorithme 3.1 calcul du pgcd selon Euclide

Entrée: deux entiers $a_0, b_0 \in \mathbb{Z}$
Sortie: le pgcd positif de a_0 et b_0

$a \leftarrow a_0, \quad b \leftarrow b_0$ // invariant $\operatorname{pgcd}(a,b) = \operatorname{pgcd}(a_0,b_0)$
tant que $b \neq 0$ **faire**
  $r \leftarrow a \operatorname{rem} b$ // $a = qb + r$ et $0 \leq r < |b|$
  $a \leftarrow b, \quad b \leftarrow r$ // $\operatorname{pgcd}(a,b)$ reste invariant
fin tant que
retourner $|a|$ // $\operatorname{pgcd}(a,0) = |a|$

L'algorithme d'Euclide, prêt à programmer

Algorithme 3.2 calcul du pgcd selon Euclide

Entrée: deux entiers $a_0, b_0 \in \mathbb{Z}$
Sortie: le pgcd positif de a_0 et b_0

$a \leftarrow a_0,\quad b \leftarrow b_0$	// invariant $\operatorname{pgcd}(a,b) = \operatorname{pgcd}(a_0,b_0)$
tant que $b \neq 0$ **faire**	
$\quad r \leftarrow a \operatorname{rem} b$	// $a = qb + r$ et $0 \leq r < \lvert b \rvert$
$\quad a \leftarrow b,\quad b \leftarrow r$	// $\operatorname{pgcd}(a,b)$ reste invariant
fin tant que	
retourner $\lvert a \rvert$	// $\operatorname{pgcd}(a,0) = \lvert a \rvert$

Théorème

L'algorithme d'Euclide explicité ci-dessus est correct :

- *Il se termine après un nombre fini d'itérations (au plus $|b_0|$).*
- *Il renvoie le pgcd positif de a_0 et b_0 comme spécifié.*

L’algorithme d’Euclide : preuve de correction

Preuve de correction. Il faut montrer la terminaison de l’algorithme puis la correction du résultat renvoyé.

L’algorithme d’Euclide : preuve de correction

Preuve de correction. Il faut montrer la terminaison de l’algorithme puis la correction du résultat renvoyé.

Terminaison : Tant que $b \neq 0$, chaque itération diminue la valeur absolue $|b|$.
Or, dans $\mathbb{N}$ il n’existe pas de suite décroissante infinie.
On arrive donc à $b = 0$ et l’algorithme s’arrête.

L'algorithme d'Euclide : preuve de correction

Preuve de correction. Il faut montrer la terminaison de l'algorithme puis la correction du résultat renvoyé.

Terminaison : Tant que $b \neq 0$, chaque itération diminue la valeur absolue $|b|$.
Or, dans $\mathbb{N}$ il n'existe pas de suite décroissante infinie.
On arrive donc à $b = 0$ et l'algorithme s'arrête.

Correction : On cherche à calculer la valeur $\operatorname{pgcd}(a_0, a_0)$.
Initialement $a = a_0$ et $b = b_0$ donc $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(a_0, b_0)$.

L’algorithme d’Euclide : preuve de correction

Preuve de correction. Il faut montrer la terminaison de l’algorithme puis la correction du résultat renvoyé.

Terminaison : Tant que $b \neq 0$, chaque itération diminue la valeur absolue $|b|$.
Or, dans $\mathbb{N}$ il n’existe pas de suite décroissante infinie.
On arrive donc à $b = 0$ et l’algorithme s’arrête.

Correction : On cherche à calculer la valeur $\operatorname{pgcd}(a_0, a_0)$.
Initialement $a = a_0$ et $b = b_0$ donc $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(a_0, b_0)$.

Dans chaque itération les valeurs a et b changent,
mais $\operatorname{pgcd}(a, b)$ est invariant car $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a - qb)$.

L'algorithme d'Euclide : preuve de correction

Preuve de correction. Il faut montrer la terminaison de l'algorithme puis la correction du résultat renvoyé.

Terminaison : Tant que $b \neq 0$, chaque itération diminue la valeur absolue $|b|$.
Or, dans $\mathbb{N}$ il n'existe pas de suite décroissante infinie.
On arrive donc à $b = 0$ et l'algorithme s'arrête.

Correction : On cherche à calculer la valeur $\operatorname{pgcd}(a_0, a_0)$.
Initialement $a = a_0$ et $b = b_0$ donc $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(a_0, b_0)$.

Dans chaque itération les valeurs a et b changent,
mais $\operatorname{pgcd}(a, b)$ est invariant car $\operatorname{pgcd}(a, b) = \operatorname{pgcd}(b, a - qb)$.

Si finalement $b = 0$, alors l'algorithme s'arrête
et renvoie $|a| = \operatorname{pgcd}(a, 0) = \operatorname{pgcd}(a, b) = \operatorname{pgcd}(a_0, b_0)$. $\square$

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$*, alors l'algorithme d'Euclide effectue au plus* $2n$ *itérations.*

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$*, alors l'algorithme d'Euclide effectue au plus* $2n$ *itérations.*

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$*, alors l'algorithme d'Euclide effectue au plus* $2n$ *itérations.*

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

- Si $0 < r_2 \leq \frac{1}{2} b_0$, alors $0 \leq r_3 < r_2 \leq \frac{1}{2} b_0$.

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2}b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

- Si $0 < r_2 \leq \frac{1}{2}b_0$, alors $0 \leq r_3 < r_2 \leq \frac{1}{2}b_0$.
- Si $\frac{1}{2}b_0 < r_2 < b_0$, alors $b_0 = 1 \cdot r_2 + r_3$ où $r_3 < \frac{1}{2}b_0 < r_2$.

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

- Si $0 < r_2 \leq \frac{1}{2} b_0$, alors $0 \leq r_3 < r_2 \leq \frac{1}{2} b_0$.
- Si $\frac{1}{2} b_0 < r_2 < b_0$, alors $b_0 = 1 \cdot r_2 + r_3$ où $r_3 < \frac{1}{2} b_0 < r_2$.

On conclut par récurrence : pour (r_2, r_3) où $0 \leq r_3 < 2^{n-1}$ l'algorithme nécessite au plus $2(n-1)$ itérations, donc au plus $2n$ pour (a_0, b_0). □

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$*, alors l'algorithme d'Euclide effectue au plus* $2n$ *itérations.*

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2} b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

- Si $0 < r_2 \leq \frac{1}{2} b_0$, alors $0 \leq r_3 < r_2 \leq \frac{1}{2} b_0$.
- Si $\frac{1}{2} b_0 < r_2 < b_0$, alors $b_0 = 1 \cdot r_2 + r_3$ où $r_3 < \frac{1}{2} b_0 < r_2$.

On conclut par récurrence : pour (r_2, r_3) où $0 \leq r_3 < 2^{n-1}$ l'algorithme nécessite au plus $2(n-1)$ itérations, donc au plus $2n$ pour (a_0, b_0). □

Corollaire

Pour $\operatorname{len}(a_0), \operatorname{len}(b_0) \leq n$ *l'algorithme d'Euclide est de complexité* $\tilde{O}(n^2)$*.*

L'algorithme d'Euclide : complexité

Proposition

Si $0 \leq b_0 < 2^n$, alors l'algorithme d'Euclide effectue au plus $2n$ itérations.

Démonstration. Si $n = 0$, alors $b_0 = 0$ et l'algorithme s'arrête.

Si $n \geq 1$, après au plus deux itérations on a $0 \leq b \leq \frac{1}{2}b_0 < 2^{n-1}$:

Posons $r_0 = a_0$ et $r_1 = b_0$ puis $r_2 := r_0 \operatorname{rem} r_1$.

- Si $r_2 = 0$, l'algorithme s'arrête après une itération.

Sinon, regardons l'étape suivante, $r_3 := r_1 \operatorname{rem} r_2$.

- Si $0 < r_2 \leq \frac{1}{2}b_0$, alors $0 \leq r_3 < r_2 \leq \frac{1}{2}b_0$.
- Si $\frac{1}{2}b_0 < r_2 < b_0$, alors $b_0 = 1 \cdot r_2 + r_3$ où $r_3 < \frac{1}{2}b_0 < r_2$.

On conclut par récurrence : pour (r_2, r_3) où $0 \leq r_3 < 2^{n-1}$ l'algorithme nécessite au plus $2(n-1)$ itérations, donc au plus $2n$ pour (a_0, b_0). □

Corollaire

Pour $\operatorname{len}(a_0), \operatorname{len}(b_0) \leq n$ l'algorithme d'Euclide est de complexité $\tilde{O}(n^2)$.

Remarque. On peut calculer le pgcd en $\tilde{O}(n)$, voir Gathen–Gerhard §11.1.

Le théorème de Bézout

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ *il existe des coefficients* $u, v \in \mathbb{Z}$ *(en général non uniques) tels que* $\operatorname{pgcd}(a, b) = au + bv$*.*

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ *il existe des coefficients* $u, v \in \mathbb{Z}$ *(en général non uniques) tels que* $\operatorname{pgcd}(a, b) = au + bv$*.*

Démonstration. Reconsidérons l'algorithme d'Euclide :

$$
\begin{array}{rclcrl}
r_0 \leftarrow a & & & & & \\
r_1 \leftarrow b & & & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & & 0 \le r_2 < |r_1| \\
\dots & & \dots & & \dots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & & 0 \le r_n < |r_{n-1}| \\
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n & \text{càd} & r_{n-1} = r_n q_n + r_{n+1}, & & r_{n+1} = 0
\end{array}
$$

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ il existe des coefficients $u, v \in \mathbb{Z}$ (en général non uniques) tels que $\operatorname{pgcd}(a, b) = au + bv$.

Démonstration. Reconsidérons l'algorithme d'Euclide :

$$\begin{array}{rlcll}
r_0 \leftarrow a & & & & \\
r_1 \leftarrow b & & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & & 0 \le r_2 < |r_1| \\
\dots & & \dots & & \dots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & & 0 \le r_n < |r_{n-1}| \\
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n & \text{càd} & r_{n-1} = r_n q_n + r_{n+1}, & & r_{n+1} = 0
\end{array}$$

On a $\operatorname{pgcd}(a, b) = |r_n| = r_n u_n + r_{n+1} v_n$ où $u_n \in \{\pm 1\}$ et $v_n \in \mathbb{Z}$.

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ il existe des coefficients $u, v \in \mathbb{Z}$ (en général non uniques) tels que $\operatorname{pgcd}(a, b) = au + bv$.

Démonstration. Reconsidérons l'algorithme d'Euclide :

$$\begin{array}{rlcll}
r_0 & \leftarrow a & & & \\
r_1 & \leftarrow b & & & \\
r_2 & \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
 & \ldots & & \ldots & \ldots \\
r_n & \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}| \\
r_{n+1} & \leftarrow r_{n-1} \operatorname{rem} r_n & \text{càd} & r_{n-1} = r_n q_n + r_{n+1}, & r_{n+1} = 0
\end{array}$$

On a $\operatorname{pgcd}(a, b) = |r_n| = r_n u_n + r_{n+1} v_n$ où $u_n \in \{\pm 1\}$ et $v_n \in \mathbb{Z}$.

Dans $\operatorname{pgcd}(a, b) = r_k u_k + r_{k+1} v_k$ on substitue $r_{k+1} = r_{k-1} - r_k q_k$:

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ il existe des coefficients $u, v \in \mathbb{Z}$ (en général non uniques) tels que $\operatorname{pgcd}(a, b) = au + bv$.

Démonstration. Reconsidérons l'algorithme d'Euclide :

$$\begin{array}{rlcll}
r_0 \leftarrow a & & & & \\
r_1 \leftarrow b & & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & & 0 \le r_2 < |r_1| \\
\dots & & \dots & & \dots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & & 0 \le r_n < |r_{n-1}| \\
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n & \text{càd} & r_{n-1} = r_n q_n + r_{n+1}, & & r_{n+1} = 0
\end{array}$$

On a $\operatorname{pgcd}(a, b) = |r_n| = r_n u_n + r_{n+1} v_n$ où $u_n \in \{\pm 1\}$ et $v_n \in \mathbb{Z}$.

Dans $\operatorname{pgcd}(a, b) = r_k u_k + r_{k+1} v_k$ on substitue $r_{k+1} = r_{k-1} - r_k q_k$:

$$\operatorname{pgcd}(a, b) = r_k u_k + (r_{k-1} - r_k q_k) v_k = r_{k-1} \underbrace{v_k}_{u_{k-1}} + r_k \underbrace{(u_k - q_k v_k)}_{v_{k-1}}.$$

Théorème (identité de Bézout)

Pour toute paire $a, b \in \mathbb{Z}$ *il existe des coefficients* $u, v \in \mathbb{Z}$ *(en général non uniques) tels que* $\operatorname{pgcd}(a, b) = au + bv$*.*

Démonstration. Reconsidérons l'algorithme d'Euclide :

$$\begin{array}{rlcrr}
r_0 \leftarrow a & & & & \\
r_1 \leftarrow b & & & & \\
r_2 \leftarrow r_0 \operatorname{rem} r_1 & \text{càd} & r_0 = r_1 q_1 + r_2, & 0 \leq r_2 < |r_1| \\
\dots & & \dots & \dots \\
r_n \leftarrow r_{n-2} \operatorname{rem} r_{n-1} & \text{càd} & r_{n-2} = r_{n-1} q_{n-1} + r_n, & 0 \leq r_n < |r_{n-1}| \\
r_{n+1} \leftarrow r_{n-1} \operatorname{rem} r_n & \text{càd} & r_{n-1} = r_n q_n + r_{n+1}, & r_{n+1} = 0
\end{array}$$

On a $\operatorname{pgcd}(a, b) = |r_n| = r_n u_n + r_{n+1} v_n$ où $u_n \in \{\pm 1\}$ et $v_n \in \mathbb{Z}$.

Dans $\operatorname{pgcd}(a, b) = r_k u_k + r_{k+1} v_k$ on substitue $r_{k+1} = r_{k-1} - r_k q_k$:

$$\operatorname{pgcd}(a, b) = r_k u_k + (r_{k-1} - r_k q_k) v_k = r_{k-1} \underbrace{v_k}_{u_{k-1}} + r_k \underbrace{(u_k - q_k v_k)}_{v_{k-1}}.$$

En remontant on trouve finalement $\operatorname{pgcd}(a, b) = a u_0 + b v_0$. $\square$

L’algorithme d’Euclide–Bézout, prêt à programmer

La description précédente ne se prête pas bien à la programmation.

La description précédente ne se prête pas bien à la programmation.
Voici une version prête à programmer :

Algorithme 3.4 l'algorithme d'Euclide–Bézout

Entrée: deux entiers $a_0, b_0 \in \mathbb{Z}$
Sortie: trois entiers d, u, v tels que $d = a_0 u + b_0 v$ soit le pgcd positif de a_0 et b_0.

$\begin{pmatrix} a & u & v \\ b & s & t \end{pmatrix} \leftarrow \begin{pmatrix} a_0 & 1 & 0 \\ b_0 & 0 & 1 \end{pmatrix}$ // invariant $\begin{cases} a = a_0 u + b_0 v \\ b = a_0 s + b_0 t \end{cases}$

tant que $b \neq 0$ **faire**

$q \leftarrow a \operatorname{quo} b, \quad \begin{pmatrix} a & u & v \\ b & s & t \end{pmatrix} \leftarrow \begin{pmatrix} b & s & t \\ a - qb & u - qs & v - qt \end{pmatrix}$

fin tant que

si $a < 0$ **alors** $(a, u, v) \leftarrow -(a, u, v)$

retourner le triplet (a, u, v)

L'algorithme d'Euclide–Bézout, prêt à programmer

La description précédente ne se prête pas bien à la programmation.
Voici une version prête à programmer :

Algorithme 3.5 l'algorithme d'Euclide–Bézout

Entrée: deux entiers $a_0, b_0 \in \mathbb{Z}$
Sortie: trois entiers d, u, v tels que $d = a_0 u + b_0 v$ soit le pgcd positif de a_0 et b_0.

$\begin{pmatrix} a & u & v \\ b & s & t \end{pmatrix} \leftarrow \begin{pmatrix} a_0 & 1 & 0 \\ b_0 & 0 & 1 \end{pmatrix}$ // invariant $\begin{cases} a = a_0 u + b_0 v \\ b = a_0 s + b_0 t \end{cases}$

tant que $b \neq 0$ **faire**
  $q \leftarrow a \operatorname{quo} b,\quad \begin{pmatrix} a & u & v \\ b & s & t \end{pmatrix} \leftarrow \begin{pmatrix} b & s & t \\ a - qb & u - qs & v - qt \end{pmatrix}$
fin tant que
si $a < 0$ **alors** $(a, u, v) \leftarrow -(a, u, v)$
retourner le triplet (a, u, v)

Exercice

Prouver que l'algorithme 3.3 est correct et majorer le temps de calcul.

Sommaire

Entiers premiers entre eux

Définition

On dit que $a, b \in \mathbb{Z}$ sont *premiers entre eux* si $\operatorname{pgcd}(a, b) = 1$.

Entiers premiers entre eux

Définition

On dit que $a, b \in \mathbb{Z}$ sont *premiers entre eux* si $\mathrm{pgcd}(a, b) = 1$.

Proposition

Deux entiers $a, b \in \mathbb{Z}$ sont premiers entre eux si et seulement s'il existe $u, v \in \mathbb{Z}$ tels que $au + bv = 1$.

Définition

On dit que $a, b \in \mathbb{Z}$ sont *premiers entre eux* si $\mathrm{pgcd}(a,b) = 1$.

Proposition

Deux entiers $a, b \in \mathbb{Z}$ *sont premiers entre eux si et seulement s'il existe* $u, v \in \mathbb{Z}$ *tels que* $au + bv = 1$*.*

Démonstration.
« $\Rightarrow$ » Si $\mathrm{pgcd}(a,b) = 1$ alors le théorème de Bézout assure l'existence de coefficients $u, v \in \mathbb{Z}$ tels que $au + bv = 1$.

Définition

On dit que $a, b \in \mathbb{Z}$ sont *premiers entre eux* si $\mathrm{pgcd}(a,b) = 1$.

Proposition

Deux entiers $a, b \in \mathbb{Z}$ sont premiers entre eux si et seulement s'il existe $u, v \in \mathbb{Z}$ tels que $au + bv = 1$.

Démonstration.
« $\Rightarrow$ » Si $\mathrm{pgcd}(a,b) = 1$ alors le théorème de Bézout assure l'existence de coefficients $u, v \in \mathbb{Z}$ tels que $au + bv = 1$.

« $\Leftarrow$ » Si $d \mid a$ et $d \mid b$ alors $d \mid au$ et $d \mid bv$, puis $d \mid au + bv$.
Or, $d \mid 1$ implique $d = \pm 1$, donc $\mathrm{pgcd}(a,b) = 1$. $\square$

Les lemmes de Gauss et d’Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Les lemmes de Gauss et d'Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$. *Si* $\mathrm{pgcd}(a, b) = 1$, *alors* $a \mid bc$ *implique* $a \mid c$.

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\operatorname{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\operatorname{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.

Les lemmes de Gauss et d'Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\operatorname{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\operatorname{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.
La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\operatorname{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\operatorname{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.
La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.
On trouve $a(uc + a'v) = auc + bcv = (au + bv)c = c$ d'où $a \mid c$. $\square$

Les lemmes de Gauss et d'Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\operatorname{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\operatorname{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.
La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.
On trouve $a(uc + a'v) = auc + bcv = (au + bv)c = c$ d'où $a \mid c$. □

Lemme (d'Euclide)

Si p *est un nombre premier, alors* $p \mid ab$ *implique* $p \mid a$ *ou* $p \mid b$*.*

Les lemmes de Gauss et d'Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\mathrm{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\mathrm{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.

La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.

On trouve $a(uc + a'v) = auc + bcv = (au + bv)c = c$ d'où $a \mid c$. $\square$

Lemme (d'Euclide)

Si p *est un nombre premier, alors* $p \mid ab$ *implique* $p \mid a$ *ou* $p \mid b$*.*

Démonstration. Pour $d = \mathrm{pgcd}(p, a)$ on a $d \mid p$, donc $d = 1$ ou $d = p$.

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$. Si $\operatorname{pgcd}(a, b) = 1$, alors $a \mid bc$ implique $a \mid c$.

Démonstration. Si $\operatorname{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.
La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.
On trouve $a(uc + a'v) = auc + bcv = (au + bv)c = c$ d'où $a \mid c$. □

Lemme (d'Euclide)

Si p est un nombre premier, alors $p \mid ab$ implique $p \mid a$ ou $p \mid b$.

Démonstration. Pour $d = \operatorname{pgcd}(p, a)$ on a $d \mid p$, donc $d = 1$ ou $d = p$.
Si $d = 1$, alors $p \mid b$ par le lemme de Gauss.

Les lemmes de Gauss et d'Euclide

Les deux résultats fondamentaux suivants sont souvent utiles.

Lemme (de Gauss)

Soient $a, b, c \in \mathbb{Z}$*. Si* $\mathrm{pgcd}(a, b) = 1$*, alors* $a \mid bc$ *implique* $a \mid c$*.*

Démonstration. Si $\mathrm{pgcd}(a, b) = 1$, on a $au + bv = 1$ d'après Bézout.
La divisibilité $a \mid bc$ veut dire qu'il existe $a' \in \mathbb{Z}$ tel que $aa' = bc$.
On trouve $a(uc + a'v) = auc + bcv = (au + bv)c = c$ d'où $a \mid c$. $\square$

Lemme (d'Euclide)

Si p *est un nombre premier, alors* $p \mid ab$ *implique* $p \mid a$ *ou* $p \mid b$*.*

Démonstration. Pour $d = \mathrm{pgcd}(p, a)$ on a $d \mid p$, donc $d = 1$ ou $d = p$.
Si $d = 1$, alors $p \mid b$ par le lemme de Gauss. Si $d = p$, alors $p \mid a$. $\square$

Théorème fondamental de l’arithmétique

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ *s’écrit de manière unique comme produit* $n = p_1 p_2 \cdots p_\ell$ *de nombres premiers* $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$.

Théorème fondamental de l'arithmétique

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ *s'écrit de manière unique comme produit* $n = p_1 p_2 \cdots p_\ell$ *de nombres premiers* $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$*.*

Encore plus que l'existence c'est l'unicité qui est remarquable !

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ *s'écrit de manière unique comme produit* $n = p_1 p_2 \cdots p_\ell$ *de nombres premiers* $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$*.*

Encore plus que l'existence c'est l'unicité qui est remarquable !

Explicitons donc l'énoncé d'unicité : Si $n = p_1 \cdots p_\ell$ et $n = q_1 \cdots q_k$ sont deux décompositions en facteurs premiers telles que $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$ et $1 < q_1 \leq q_2 \leq \cdots \leq q_k$, alors $\ell = k$ et $p_1 = q_1, \ldots, p_\ell = q_\ell$.

Théorème fondamental de l'arithmétique

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ s'écrit de manière unique comme produit $n = p_1 p_2 \cdots p_\ell$ de nombres premiers $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$.

Encore plus que l'existence c'est l'unicité qui est remarquable !

Explicitons donc l'énoncé d'unicité : Si $n = p_1 \cdots p_\ell$ et $n = q_1 \cdots q_k$ sont deux décompositions en facteurs premiers telles que $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$ et $1 < q_1 \leq q_2 \leq \cdots \leq q_k$, alors $\ell = k$ et $p_1 = q_1, \ldots, p_\ell = q_\ell$.

Exemple

On peut décomposer $n = 60$ de différentes manières :

$$60 = 6 \cdot 10 = (-4) \cdot 3 \cdot (-5) = \ldots$$

Théorème fondamental de l'arithmétique

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ *s'écrit de manière unique comme produit* $n = p_1 p_2 \cdots p_\ell$ *de nombres premiers* $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$*.*

Encore plus que l'existence c'est l'unicité qui est remarquable !

Explicitons donc l'énoncé d'unicité : Si $n = p_1 \cdots p_\ell$ et $n = q_1 \cdots q_k$ sont deux décompositions en facteurs premiers telles que $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$ et $1 < q_1 \leq q_2 \leq \cdots \leq q_k$, alors $\ell = k$ et $p_1 = q_1, \ldots, p_\ell = q_\ell$.

Exemple

On peut décomposer $n = 60$ de différentes manières :

$$60 = 6 \cdot 10 = (-4) \cdot 3 \cdot (-5) = \ldots$$

Si l'on exige des facteurs premiers, il reste encore des ambiguïtés :

$$60 = (-2) \cdot 2 \cdot (-3) \cdot 5 = 3 \cdot 2 \cdot (-5) \cdot (-2) = \ldots$$

Théorème fondamental de l'arithmétique

Théorème (décomposition en facteurs premiers)

Tout entier $n > 1$ *s'écrit de manière unique comme produit* $n = p_1 p_2 \cdots p_\ell$ *de nombres premiers* $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$*.*

Encore plus que l'existence c'est l'unicité qui est remarquable !

Explicitons donc l'énoncé d'unicité : Si $n = p_1 \cdots p_\ell$ et $n = q_1 \cdots q_k$ sont deux décompositions en facteurs premiers telles que $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$ et $1 < q_1 \leq q_2 \leq \cdots \leq q_k$, alors $\ell = k$ et $p_1 = q_1, \ldots, p_\ell = q_\ell$.

Exemple

On peut décomposer $n = 60$ de différentes manières :

$$60 = 6 \cdot 10 = (-4) \cdot 3 \cdot (-5) = \ldots$$

Si l'on exige des facteurs premiers, il reste encore des ambiguïtés :

$$60 = (-2) \cdot 2 \cdot (-3) \cdot 5 = 3 \cdot 2 \cdot (-5) \cdot (-2) = \ldots$$

Si l'on exige en plus que $1 < p_1 \leq p_2 \leq \cdots \leq p_\ell$, il ne reste que

$$60 = 2 \cdot 2 \cdot 3 \cdot 5.$$

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.
3. Si n est composé, trouver sa décomposition en facteurs premiers.

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.
3. Si n est composé, trouver sa décomposition en facteurs premiers.

Exercice

Expliciter les méthodes évidentes et estimer leur coût. Combien de temps faut-il dans le pire cas pour $n \approx 10^{20}$? pour $n \approx 10^{40}$? pour $n \approx 10^{60}$?

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.
3. Si n est composé, trouver sa décomposition en facteurs premiers.

Exercice

Expliciter les méthodes évidentes et estimer leur coût. Combien de temps faut-il dans le pire cas pour $n \approx 10^{20}$? pour $n \approx 10^{40}$? pour $n \approx 10^{60}$?

Conclusion. Pour les entiers de taille modeste, les méthodes naïves sont suffisamment efficaces.

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.
3. Si n est composé, trouver sa décomposition en facteurs premiers.

Exercice

Expliciter les méthodes évidentes et estimer leur coût. Combien de temps faut-il dans le pire cas pour $n \approx 10^{20}$? pour $n \approx 10^{40}$? pour $n \approx 10^{60}$?

Conclusion. Pour les entiers de taille modeste, les méthodes naïves sont suffisamment efficaces. Pour les grands entiers, par contre, elles échouent !

Problèmes algorithmiques

Étant donné un entier n, trois problèmes pratiques se posent :

1. Déterminer rapidement si n est premier ou composé.
2. Si n est premier, en trouver une preuve concise et facilement vérifiable.
3. Si n est composé, trouver sa décomposition en facteurs premiers.

Exercice

Expliciter les méthodes évidentes et estimer leur coût. Combien de temps faut-il dans le pire cas pour $n \approx 10^{20}$? pour $n \approx 10^{40}$? pour $n \approx 10^{60}$?

Conclusion. Pour les entiers de taille modeste, les méthodes naïves sont suffisamment efficaces. Pour les grands entiers, par contre, elles échouent !

On discutera plus tard des méthodes plus efficaces.

Sommaire

Le pire cas de l'algorithme d'Euclide

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Le pire cas de l'algorithme d'Euclide

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Exercice

1. Montrer que l'algorithme d'Euclide nécessite exactement k itérations pour calculer $\mathrm{pgcd}(f_{k+1}, f_k)$ où $k \geq 1$.

Le pire cas de l'algorithme d'Euclide

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Exercice

1. Montrer que l'algorithme d'Euclide nécessite exactement k itérations pour calculer $\mathrm{pgcd}(f_{k+1}, f_k)$ où $k \geq 1$.
2. Réciproquement, si l'algorithme d'Euclide nécessite k itérations pour calculer $\mathrm{pgcd}(a, b)$ où $a > b > 0$, alors $a \geq f_{k+1}$ et $b \geq f_k$.

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Exercice

1. Montrer que l'algorithme d'Euclide nécessite exactement k itérations pour calculer $\mathrm{pgcd}(f_{k+1}, f_k)$ où $k \geq 1$.
2. Réciproquement, si l'algorithme d'Euclide nécessite k itérations pour calculer $\mathrm{pgcd}(a, b)$ où $a > b > 0$, alors $a \geq f_{k+1}$ et $b \geq f_k$.
3. Montrer que $f_k = (\lambda_+^{k+1} - \lambda_-^{k+1})/\sqrt{5}$ où $\lambda_\pm = (1 \pm \sqrt{5})/2$.
   Pour information : on a $\sqrt{5} \approx 2.24$ donc $\lambda_+ \approx 1.62$ et $\lambda_- \approx -0.62$.

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

Exercice

1. Montrer que l'algorithme d'Euclide nécessite exactement k itérations pour calculer $\operatorname{pgcd}(f_{k+1}, f_k)$ où $k \geq 1$.
2. Réciproquement, si l'algorithme d'Euclide nécessite k itérations pour calculer $\operatorname{pgcd}(a, b)$ où $a > b > 0$, alors $a \geq f_{k+1}$ et $b \geq f_k$.
3. Montrer que $f_k = (\lambda_+^{k+1} - \lambda_-^{k+1})/\sqrt{5}$ où $\lambda_\pm = (1 \pm \sqrt{5})/2$.
   Pour information : on a $\sqrt{5} \approx 2.24$ donc $\lambda_+ \approx 1.62$ et $\lambda_- \approx -0.62$.
4. Pour $|b| < 2^n$, dans le pire cas l'algorithme d'Euclide calcule $\operatorname{pgcd}(a, b)$ avec $k \sim \frac{n}{\log_2(\lambda_+)}$ itérations, soit environ $1.44n$ itérations.

Le pire cas de l'algorithme d'Euclide

Définition

La suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ est définie par $f_0 = 1$ et $f_1 = 1$
puis par la formule de récurrence $f_{k+2} = f_{k+1} + f_k$ pour tout $k \geq 0$.

Pour illustration, les premiers termes sont $1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$

Exercice

1. Montrer que l'algorithme d'Euclide nécessite exactement k itérations pour calculer $\operatorname{pgcd}(f_{k+1}, f_k)$ où $k \geq 1$.
2. Réciproquement, si l'algorithme d'Euclide nécessite k itérations pour calculer $\operatorname{pgcd}(a, b)$ où $a > b > 0$, alors $a \geq f_{k+1}$ et $b \geq f_k$.
3. Montrer que $f_k = (\lambda_+^{k+1} - \lambda_-^{k+1})/\sqrt{5}$ où $\lambda_\pm = (1 \pm \sqrt{5})/2$.
   Pour information : on a $\sqrt{5} \approx 2.24$ donc $\lambda_+ \approx 1.62$ et $\lambda_- \approx -0.62$.
4. Pour $|b| < 2^n$, dans le pire cas l'algorithme d'Euclide calcule $\operatorname{pgcd}(a, b)$ avec $k \sim \frac{n}{\log_2(\lambda_+)}$ itérations, soit environ $1.44n$ itérations.

On peut aussi s'intéresser au nombre moyen d'itérations.
Pour plus d'information voir Knuth, vol. 2, §4.5.3.

Non-unicité des coefficients de Bézout

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a, b) = au_0 + bv_0$.

Non-unicité des coefficients de Bézout

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$ vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$ vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a,b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u,v) \in \mathbb{Z}^2$ vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Réciproquement on veut montrer que l'identité $au + bv = d$ implique que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ pour un certain $k \in \mathbb{Z}$.

Non-unicité des coefficients de Bézout

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$ vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Réciproquement on veut montrer que l'identité $au + bv = d$ implique que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ pour un certain $k \in \mathbb{Z}$.

On pose $a_0 = a/d$ et $b_0 = b/d$ pour avoir $a_0u_0 + b_0v_0 = 1$ et $a_0u + b_0v = 1$.

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \operatorname{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$ vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Réciproquement on veut montrer que l'identité $au + bv = d$ implique que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ pour un certain $k \in \mathbb{Z}$.

On pose $a_0 = a/d$ et $b_0 = b/d$ pour avoir $a_0u_0 + b_0v_0 = 1$ et $a_0u + b_0v = 1$.
La différence $u_1 := u - u_0$ et $v_1 := v - v_0$ vérifie alors $a_0u_1 + b_0v_1 = 0$.

Non-unicité des coefficients de Bézout

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \mathrm{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$
vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Réciproquement on veut montrer que l'identité $au + bv = d$
implique que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ pour un certain $k \in \mathbb{Z}$.

On pose $a_0 = a/d$ et $b_0 = b/d$ pour avoir $a_0u_0 + b_0v_0 = 1$ et $a_0u + b_0v = 1$.
La différence $u_1 := u - u_0$ et $v_1 := v - v_0$ vérifie alors $a_0u_1 + b_0v_1 = 0$.
Or, $\mathrm{pgcd}(a_0, b_0) = 1$ et $a_0 \mid b_0v_1$ implique $a_0 \mid v_1$ d'après Gauss.

Non-unicité des coefficients de Bézout

Exercice

Soient $a, b \in \mathbb{Z}$ deux entiers et soit $d = \operatorname{pgcd}(a, b) = au_0 + bv_0$.
Montrer que l'ensemble des coefficients de Bézout $(u, v) \in \mathbb{Z}^2$
vérifiant $au + bv = d$ est donné par $\{(u_0 + k\frac{b}{d}, v_0 - k\frac{a}{d}) \mid k \in \mathbb{Z}\}$

Solution.
On constate que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ vérifient $au + bv = d$.

Réciproquement on veut montrer que l'identité $au + bv = d$
implique que $u = u_0 + k\frac{b}{d}$ et $v = v_0 - k\frac{a}{d}$ pour un certain $k \in \mathbb{Z}$.

On pose $a_0 = a/d$ et $b_0 = b/d$ pour avoir $a_0u_0 + b_0v_0 = 1$ et $a_0u + b_0v = 1$.
La différence $u_1 := u - u_0$ et $v_1 := v - v_0$ vérifie alors $a_0u_1 + b_0v_1 = 0$.
Or, $\operatorname{pgcd}(a_0, b_0) = 1$ et $a_0 \mid b_0v_1$ implique $a_0 \mid v_1$ d'après Gauss.
On conclut que $v_1 = ka_0$, puis $u_1 = -kb_0$. ☺

Factorielle et coefficient binomial

Pour tout nombre naturel $n \in \mathbb{N}$ on définit la *factorielle* $n!$ par récurrence, en posant $0! := 1$ et $n! := (n-1)! \cdot n$ si $n \geq 1$. Pour $0 \leq k \leq n$ on définit le *coefficient binomial* par $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, et $\binom{n}{k} := 0$ si $k < 0$ ou $k > n$.

Factorielle et coefficient binomial

Pour tout nombre naturel $n \in \mathbb{N}$ on définit la *factorielle* $n!$ par récurrence, en posant $0! := 1$ et $n! := (n-1)! \cdot n$ si $n \geq 1$. Pour $0 \leq k \leq n$ on définit le *coefficient binomial* par $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, et $\binom{n}{k} := 0$ si $k < 0$ ou $k > n$.

Exercice

Montrer la règle du triangle de Pascal $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.
En déduire que $\binom{n}{k}$ est en fait un entier pour tout n, k.

Factorielle et coefficient binomial

Pour tout nombre naturel $n \in \mathbb{N}$ on définit la *factorielle* $n!$ par récurrence, en posant $0! := 1$ et $n! := (n-1)! \cdot n$ si $n \geq 1$. Pour $0 \leq k \leq n$ on définit le *coefficient binomial* par $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, et $\binom{n}{k} := 0$ si $k < 0$ ou $k > n$.

Exercice

Montrer la règle du triangle de Pascal $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.
En déduire que $\binom{n}{k}$ est en fait un entier pour tout n, k.

Exercice

Soit p un nombre premier et $0 < k < p$. Montrer que p divise $\binom{p}{k}$. (Quel résultat du cours sert ici ?) Est-ce que n divise $\binom{n}{k}$ si n n'est pas premier ?

Factorielle et coefficient binomial

Pour tout nombre naturel $n \in \mathbb{N}$ on définit la *factorielle* $n!$ par récurrence, en posant $0! := 1$ et $n! := (n-1)! \cdot n$ si $n \geq 1$. Pour $0 \leq k \leq n$ on définit le *coefficient binomial* par $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, et $\binom{n}{k} := 0$ si $k < 0$ ou $k > n$.

Exercice

Montrer la règle du triangle de Pascal $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.
En déduire que $\binom{n}{k}$ est en fait un entier pour tout n, k.

Exercice

Soit p un nombre premier et $0 < k < p$. Montrer que p divise $\binom{p}{k}$. (Quel résultat du cours sert ici ?) Est-ce que n divise $\binom{n}{k}$ si n n'est pas premier ?

Exercice

Montrer que $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$. Préciser les hypothèses implicites. Quelles règles de calcul sont utilisées ?

Factorielle et coefficient binomial

Pour tout nombre naturel $n \in \mathbb{N}$ on définit la *factorielle* $n!$ par récurrence, en posant $0! := 1$ et $n! := (n-1)! \cdot n$ si $n \geq 1$. Pour $0 \leq k \leq n$ on définit le *coefficient binomial* par $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, et $\binom{n}{k} := 0$ si $k < 0$ ou $k > n$.

Exercice

Montrer la règle du triangle de Pascal $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.
En déduire que $\binom{n}{k}$ est en fait un entier pour tout n, k.

Exercice

Soit p un nombre premier et $0 < k < p$. Montrer que p divise $\binom{p}{k}$. (Quel résultat du cours sert ici ?) Est-ce que n divise $\binom{n}{k}$ si n n'est pas premier ?

Exercice

Montrer que $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$. Préciser les hypothèses implicites. Quelles règles de calcul sont utilisées ?

Exercice

Implémenter une fonction (en C++, disons) qui calcule $\binom{n}{k}$ à partir des quatre opérations élémentaires des entiers. Essayer de le faire le plus efficacement possible pour des valeurs n et k éventuellement grandes.

Un joli théorème de Dirichlet

Un joli théorème de Dirichlet affirme que deux entiers aléatoires sont premiers entre eux avec probabilité $6/\pi^2$, soit environ 60%.

Un joli théorème de Dirichlet

Un joli théorème de Dirichlet affirme que deux entiers aléatoires sont premiers entre eux avec probabilité $6/\pi^2$, soit environ 60%.

Exercice

Heuristiquement, qu'est-ce que le théorème de Dirichlet signifie pour le calcul de $\mathrm{pgcd}(a_1, a_2, \ldots, a_n)$?

Un joli théorème de Dirichlet

Un joli théorème de Dirichlet affirme que deux entiers aléatoires sont premiers entre eux avec probabilité $6/\pi^2$, soit environ 60%.

Exercice

Heuristiquement, qu'est-ce que le théorème de Dirichlet signifie pour le calcul de $\mathrm{pgcd}(a_1, a_2, \ldots, a_n)$?

Exercice

Imaginez une « forêt mathématique » formée d'une infinité d'arbres très fins, avec un arbre planté à chaque position du réseau $\mathbb{Z}^2$ dans le plan $\mathbb{R}^2$. Vous êtes à l'origine. Quelle fraction d'arbres voyez-vous ?

Un joli théorème de Dirichlet

Un joli théorème de Dirichlet affirme que deux entiers aléatoires sont premiers entre eux avec probabilité $6/\pi^2$, soit environ 60%.

Exercice

Heuristiquement, qu'est-ce que le théorème de Dirichlet signifie pour le calcul de $\operatorname{pgcd}(a_1, a_2, \dots, a_n)$?

Exercice

Imaginez une « forêt mathématique » formée d'une infinité d'arbres très fins, avec un arbre planté à chaque position du réseau $\mathbb{Z}^2$ dans le plan $\mathbb{R}^2$. Vous êtes à l'origine. Quelle fraction d'arbres voyez-vous ?

Exercice

Pour vérifier empiriquement le théorème écrire un programme qui parcourt les paires $(a, b) \in \{1, \dots, N\}^2$ et qui compte celles vérifiant $\operatorname{pgcd}(a, b) = 1$. Que trouve-t-on pour $N = 10$? $N = 100$? $N = 1000$? $N = 10000$?

Un joli théorème de Dirichlet

Un joli théorème de Dirichlet affirme que deux entiers aléatoires sont premiers entre eux avec probabilité $6/\pi^2$, soit environ 60%.

Exercice

Heuristiquement, qu'est-ce que le théorème de Dirichlet signifie pour le calcul de $\mathrm{pgcd}(a_1, a_2, \ldots, a_n)$?

Exercice

Imaginez une « forêt mathématique » formée d'une infinité d'arbres très fins, avec un arbre planté à chaque position du réseau $\mathbb{Z}^2$ dans le plan $\mathbb{R}^2$. Vous êtes à l'origine. Quelle fraction d'arbres voyez-vous ?

Exercice

Pour vérifier empiriquement le théorème écrire un programme qui parcourt les paires $(a, b) \in \{1, \ldots, N\}^2$ et qui compte celles vérifiant $\mathrm{pgcd}(a, b) = 1$. Que trouve-t-on pour $N = 10$? $N = 100$? $N = 1000$? $N = 10000$?

Exercice

Si vous connaissez les techniques mathématiques, vous pouvez essayez d'expliciter puis de prouver ce théorème.