TD n° 6

Algorithmes probabilistes

Hugo SALOU MPI*

Dernière mise à jour le 7 novembre 2022

1 Exercice 1 : Vérification d'égalité polynomiale

1. Étant donnés deux tableaux représentant deux polynômes, on peut calculer leurs produit en concaténant ce tableau. La complexité du produit de polynômes avec cet algorithme est en $\mathcal{O}(nm)$ où n est le degré du 1er polynôme, et m est le degré du second. En effet, *dans le pire des cas*, tous les polynômes représentant les deux polynômes sont des monômes, or, la concaténation étant en $\mathcal{O}(nm)$ (pour un tableau de taille n et un de taille m). D'où la complexité en $\mathcal{O}(nm)$.
2. Afin d'évaluer ces polynômes, on utilise l'algorithme de Horner, qui est en $\mathcal{O}(n)$, donc en temps linéaire.
3. En développant ces polynômes, la complexité serait en $\mathcal{O}(n^3)$. En effet, la multiplication de deux polynômes de degrés n a une complexité en $\mathcal{O}(n^2)$. D'où la complexité en $\mathcal{O}(n^3)$ pour la multiplication de deux polynômes ayant chacun un degré n.
4. Un polynôme de degré n a, au plus, n racines. D'où, le polynôme $P - Q$, a au plus n racines (où $n = \max(\deg P, \deg Q)$). Ainsi, s'il a $n + 1$ racines, c'est alors le polynôme nul, et donc $P = Q$.

Algorithme 1 Algorithme déterministe pout tester l'égalité polynomiale en $\mathcal{O}(n^2)$

Entrée : $P = (P_i)_{i \in [\![1,m]\!]}$ et $Q = (Q_j)_{j \in [\![1,p]\!]}$ deux polynômes
$n \leftarrow \deg P$
pour $i \in [\![0, n]\!]$ **faire**
  si $P(i) \neq Q(i)$ **alors** ▷ *Avec l'algorithme de Horner, évaluation en $\mathcal{O}(n)$*
    retourner Non
retourner Oui

5.

Algorithme 2 Algorithme probabiliste pout tester l'égalité polynomiale en $\mathcal{O}(n)$

Entrée $P = (P_i)_{i \in [\![1,n]\!]}$ et $Q = (Q_j)_{j \in [\![1,n]\!]}$ deux polynômes, et $k \in \mathbb{N}$ un entier
$x \leftarrow \mathcal{U}([\![1, k \times n]\!])$
si $P(x) \neq Q(x)$ **alors**
  retourner Non
retourner Oui

Soit X la variable aléatoire de $\mathcal{U}([\![1, k \times n]\!])$. L'événement "$P \neq Q$ mais l'algorithme retourne Oui" arrive si $X \in \{j \in [\![1, kn]\!] \mid P(j) = Q(j)\} = A$. Or $|A| \leqslant n$, et $A \subseteq [\![1, kn]\!]$. Ainsi, l'événement a une probabilité de $\frac{1}{k}$.

2 Test de primalité probabiliste

2.1 Résultats mathématiques

1. — Élément neutre : soit $x \in G_n$, d'où $x \cdot 1 = 1 \times x \bmod n = x \bmod n$, et donc $1 \in G_n$ est l'élément neutre de G_n.
   — Associativité : par associativité de $\times$, et par le fait que "mod" soit une congruence, on en conclut que $\cdot$ est associative.
   — Soient $x, y \in G_n$. Ainsi, $x \cdot y = x \times y \bmod n$. Or, $x \times y \wedge n = 1$, et donc $x \cdot y \wedge n = 1$.
   — Soit $x \in G_n$, donc $x \wedge n = 1$. D'où, d'après le théorème de Bézout, il existe u et $v \in \mathbb{Z}$ deux entiers tels que $u \times x + v \times n = 1$. D'où $1 \bmod n = u \times x + v \times n \bmod n$ et donc $1 = u \times x \bmod n$. Ainsi $x^{-1} = u \in G_n$, car $u \neq 0$.
2. On sait que $1 \in E_n$. Soit $y \in E_n$, d'où $y^{n-1} \equiv 1 \; [n]$, i.e. $y \times (y^{n-2}) \equiv 1 \; [n]$, donc $y^{n-2} \in E_n$ est l'inverse de y. Soient x et $y \in E_n$. On a $(x \cdot y^{-1})^{n-1} \equiv x^{n-1} \cdot y^{n-1} \; [n] \equiv 1 \; [n]$. D'où $x \cdot y^{-1} \in E_n$. Ainsi, E_n est un sous-groupe de $(G_n, \cdot)$.
3. Soit n composé. Il existe $a \in [\![1, n-1]\!]$ tel que $a^{n-1} \not\equiv 1 \; [n]$, et donc $E_n \subsetneq G_n$. Or, *le cardinal d'un sous-groupe divise le cardinal du groupe*, et donc $|E_n| \mid |G_n| \leqslant n - 1$, donc $|E_n| \leqslant \frac{n-1}{2}$.

2.2 Algorithme

4.

Algorithme 3 Algorithme Monte-Carlo testant la primalité d'un nombre en $\mathcal{O}(k\;(\ln k)^3$

Entrée $n \in \mathbb{N}$ et $k \in \mathbb{N}$ deux entiers.

```
pour j ∈ ⟦1, k⟧ faire
    a ← 𝒰(⟦1, n − 1⟧)
    si a^(n−1) mod n ≠ 1 alors
        retourner Non
retourner Oui
```

En effet, si $|E_n| \leqslant \frac{n-1}{2}$, donc si $a \sim \mathcal{U}([\![1, n-1]\!])$, d'où $P(a \in E_n) \leqslant \frac{1}{2}$. La probabilité que l'algorithme échoue est inférieure à $\frac{1}{2^k}$.

2.3 Implémentation

Indications Pour calculer $a^b \bmod c$, on décompose b en base 2 : $b = \sum_{i=1}^{p} b_i 2^i$, et donc

$$a^b \bmod c = \Big(\prod_{i=1}^{p} a^{b_i 2^i}\Big) \bmod c = \prod_{i=0}^{p} \Big(a^{b_i 2^i} \bmod c\Big).$$

Et, $p \sim \log_2(n)$.

3 Exercice 3 : Échantillonnage

Q. 1

Algorithme 4 Échantillonnage naïf

Entrée T un tableau à n éléments, et $k \in \mathbb{N}$ avec $k \leqslant n$

```
T ← Mélanger(T)
R ← T[0..k]
retourner R
```

Q. 2 Un invariant de boucle est « $\forall p \in [\![0, I-1]\!]\,,\ P(T[p] \in \mathsf{Res}) = \frac{k}{I}$ et $\forall p \in [\![I, n]\!]\,,\ T[p] \notin \mathsf{Res}$ »

Q. 3 Notons $\underline{I}$ et $\underline{\mathsf{Res}}$ l'état des variables avant un tour de boucle ; et, $\overline{I}$ et $\overline{\mathsf{Res}}$ l'état des variables après un tour de boucle.

— Pour $k = I$, on a
   1. $\forall p \in [\![0, k-1]\!]$, $P(T[p] \in \mathsf{Res}) = 1$,
   2. $\forall p \in [\![k, n-1]\!]$, $T[p] \notin \mathsf{Res}$,
   3. $I \leqslant n$.

— Supposons $\underline{I}$, et $\underline{\mathsf{Res}}$ vérifiant l'invariant et la condition de boucle. Alors, on a
   1. $\forall p \in [\![0, \underline{I}-1]\!]$, $P(T[p] \in \underline{\mathsf{Res}}) = \frac{k}{\underline{I}}$,
   2. $\forall p \in [\![\underline{I}, n-1]\!]$, $T[p] \notin \underline{\mathsf{Res}}$,
   3. $\underline{I} < n$, la condition de boucle.

   Soit $j \in [\![0, \underline{I}]\!]$. On a $\overline{I} = \underline{I} + 1$.

Cas 1 $j < k$, et donc $\overline{\mathsf{Res}}(j) = T[\underline{I}]$, et $\forall \ell \neq j$, $\overline{\mathsf{Res}}[\ell] = \underline{\mathsf{Res}}[\ell]$.

Cas 2 $j \geqslant k$, et donc $\forall \ell$, $\overline{\mathsf{Res}}[\ell] = \underline{\mathsf{Res}}[\ell]$.

1. Soit $p \in [\![0, \underline{I}]\!]$. Montrons $P(T[p] \notin \overline{\mathsf{Res}}) = \frac{k}{\overline{I}}$. Si $p < \underline{I}$, alors

$$\begin{aligned} P(T[p] \in \overline{\mathsf{Res}}) &= P\big(T[p] \in \underline{\mathsf{Res}} \cap j \neq p\big) \\ &= \frac{k}{\underline{I}} \times \frac{\underline{I}}{\underline{I} + 1} \\ &= \frac{k}{\overline{I}}. \end{aligned}$$

Si $P = \underline{I}$, alors d'après 2. $T[p] \notin \underline{\mathsf{Res}}$, donc $P(T[p] \in \overline{\mathsf{Res}}) = P(j < k) = \frac{k}{\overline{I}+1}$.