Compléments de cours MAT309, 2022/2023

Roland Bacher

22 novembre 2023

1 La méthode de factorisation ρ de Pollard

Dans cette méthode, il y a trois ingrédients : Le théorème chinois (et l'algorithme d'Euclide pour le calcul du pgcd), le théorème des anniversaires et l'algorithme du lièvre et de la tortue (appelé l'algorithme de Floyd) pour trouver un cycle parmi les itérés $f^k(x_0)$ d'une application $f : E \longrightarrow E$ d'un ensemble fini E dans lui-même.

Théorème 1.1 (Théorème des anniversaires)**.** *L'espérance de tomber pour la prmière fois sur une boule déjà vue lors de tirages indépendants avec remise parmi N boules différentes est de l'ordre $\sqrt{N}$.*

(Lien avec les anniversaires : Dans environ une classe sur deux d'une vingtaine d'élèves ($20 \sim \sqrt{365}$) il y a deux élèves nés le même jour.)

Idée de la preuve. La probabilité que k boules tirées (uniformément, avec remise) parmi N soient toutes différentes est égale $\prod_{j=1}^{k-1}(1-j/N)$. On veut trouver une valeur k pour laquelle cette probabilité devient sensiblement plus petit que 1. On commence par prendre le logarithme qu'on remplace par son développement à l'ordre 1 (donné par $\log(1+x) \sim x$ si $|x|$ est petit) pour avoir

$$\log(\prod_{j=1}^{k-1}(1-j/N)) = \sum_{j=1}^{k-1}\log(1-j/N) \sim -\frac{1}{N}\sum_{j=1}^{k-1} j = -\frac{1}{N}\binom{k}{2} \sim -\frac{k^2}{2N}$$

qui vaut $-1/2$ pour $k = \sqrt{N}$. □

En itérant une application de la forme $x \longmapsto x^2 + a \pmod N$ (pour N un entier composé à factoriser, on s'attend tomber sur une valeur déjà vue après environ $\sqrt{N}$ itérations. L'application $x \longmapsto x^2 + a$ est cependant compatible avec le théorème chinois et elle devrait donc devenir périodique modulo p après environ $\sqrt{p}$ itérations pour p le plus petit diviseur premier de N. Si N n'est pas premier, alors $\sqrt{p}$ est au plus égal $N^{1/4}$.

Pour détecter ce cycle modulo p on utilise l'algorithme du lièvre et de la tortue : Si un lièvre court deux fois plus vite qu'une tortue dans un stade, le lièvre dépasse la tortue tous les deux tours.

Algorithme ρ de Pollard : N (entier composé) factoriser. On choisit a et x_0. On note $f(x)$ pour $x^2 + a \pmod N$.

```
x:=f(x0);
y:=f(x);
tantque pgcd(x-y,N)=1 faire
  x:=f(x); (c'est la tortue)
  y:=f(f(y)); (c'est le li\'evre qui court deux fois plus vite)
fintantque;
afficher pgcd(x-y,N);
```

L'algorithme est probabiliste : il arrive qu'il affiche N et pas un diviseur strict. Dans ce cas on relance avec un autre a et/ou un autre x_0.

L'étape la plus coûteuse est le calcul du pgcd. Concrètement, on accumule un produit de quelques termes $x - y$ et on effectue un calcul de pgcd entre le produit accumulé et l'entier N factoriser seulement de temps en temps.

L'application f de l'algo n'est pas une permutation (elle n'est jamais injective pour $N > 2$). Lorsqu'on l'itère il y a généralement un bout prépériodique avant d'entrer dans un cycle périodique. L'application f est d'ailleurs tout sauf une application aléatoire. Elle semble cependant se comporter comme une application aléatoire par rapport au théorème des anniversaire et cela suffit pour faire marcher cet algorithme (qui est probabiliste).

Attention : Remplacer l'application quadratique f par une application affine n'est pas une bonne idee. (Pourquoi ?)

Cet algorithme permet de factoriser assez facilement des entiers composés de taille environ 10^{35}.

Les algorithmes les plus rapides actuellement connus factorisent un entier de taille environ 10^{100} en environ une heure sur un PC. C'est pour cette raison qu'il faut choisir des clés RSA relativement importantes.

2 Un peu de codes correcteurs d'erreurs

Je ne vais pas essayer de faire une théorie générale mais d'illustrer la notion de code correcteur par quelques exemples de codes correcteurs linéaires sur le corps $\mathbb{F}_2$ à deux éléments.

2.1 Un code de Reed-Muller utilisé par la NASA

Sondes Mariner (1969-1973) : Transmission de photos de Mars vers la Terre (Mariner 9, 1971). Émetteurs peu puissants, grande distance, milieu interstellaire bruyant. Beaucoup d'erreurs dans la transmission des bits codant les photos prises de Mars.

On regroupe les bits par paquets de 6 pour former des éléments de l'espace vectoriel $\mathbb{F}_2^6$ de dimension 6 sur le corps $\mathbb{F}_2$ à deux éléments. On considère ensuite l'application linéaire injective de $\mathbb{F}_2^6$ dans $\mathbb{F}_2^{32}$ donnée par $m = (m_1, \ldots, m_6) \longmapsto \varphi(m) = \sum_{i=1}^{6} m_i v_i$ où

$$\begin{aligned} v_1 &= 1^{32}, \\ v_2 &= 0^{16}1^{16}, \\ v_3 &= (0^8 1^8)^2, \\ v_4 &= (0^4 1^4)^4, \\ v_5 &= (0011)^8, \\ v_6 &= (01)^{16}, \end{aligned}$$

(avec coefficients concaténés en mots de longueur 32).

Soit $m = (m_1, \ldots, m_6)$ une unité de données à transmettre. La sonde a donc émis le message $\varphi(m)$ et la terre a reçu un message corrompu $r = (r_1, \ldots, r_{32}) = \varphi(m) + e$ obtenu en additionnant à $\varphi(m)$ un vecteur d'erreurs $e = (e_1, \ldots, e_{32})$ indiquant les bits corrompues durant la transmission. La terre n'a aucune information sur le vecteur e et il faut donc essayer de récuperer le message émis $\varphi(m)$ pour retrouver le paquet de données m. Ceci sera possible si le nombre d'erreurs ne dépasse pas 7 (si le vecteur e n'a pas plus de 7 coefficients non nuls).

En effet :

Exemple concret aléatoire au tableau.

(Pour décoder $r_1, \ldots, r_{32}$ 'à la main' : regarder le nombre de couples identiques $r_{2i-1} = r_{2i}$: si ce nombre est ≥ 9, alors v_6 n'intervient pas dans $\varphi(m)$. Par contre, si $r_{2i-1} \neq r_{2i}$ pour au moins 9 valeurs de i, alors v_6 intervient dans $\varphi(m)$ (et on peut continuer avec $r + v_6$). Cela permet en plus d'identifier les ≤ 7 couples contenant une unique erreur par couple.

Après avoir ajouté v_6 si nécessaire, on peut supposer qu'au moins 9 couples r_{2i-1}, r_{2i} sont identiques. On remplace maintenant les couples identiques par leur valeur commune et les couples non identiques par un point

d'interrogation. Ceci donne un vecteur $s_1, \dots s_{16}$ à valeurs dans $\{0, 1, ?\}$. Le nombre d'erreurs restants + le nombre de ? ne dépasse pas 7 s'il n'y a pas eu plus de 7 erreurs durant la transmission. Il existe donc au moins 9 couples sans ? qui sont soit soit différents soit identiques. Ceci permet de décider de l'implication ou non de v_5. En soustrayant éventuellement le vecteur $(0, 1)^8$, on se ramène à une vecteur ayant au moins 9 paires de coordonnées consécutives égales. On remplace ce vecteur par un vecteur de taille 8 en remplaçant deux coordonnées consécutives identiques par leur valeur commun, deux points d'interrogation par un point d'interrogation et un couple comportant un seul point d'interrogation par la valeur numérique de l'autre point. Ceci donne un vecteur dans $\{0, 1, ?\}^8$ ayant au plus 3 erreurs (peut-être signalé par un point d'interrogation) : Les points d'interrogations isolé ont été négligé et les erreurs non identifié par un point d'interrogations doivent être des erreurs doubles. Il existe donc soit 5 couples différents soit 5 couples égaux (sans points d'interrogation) ce qui permet de déterminer l'implication de v_4. On additionne $(0, 1)^4$ si nécessaire et on divise la longueur par 2 en effaçant les points d'interrogations unique dans les couples et en prenant la valeur commune autrement pour se ramener à un vecteur de longueur 4 qui contient au plus un point d'interrogation et qui est sans erreurs sur les autres bits. Cela donne v_3 et permet de déterminer la valeur du point d'interrogation. On obtient ensuite v_2 et v_1.

Argument théorique :

Pourquoi peut-on toujours corriger ≤ 7 erreurs en utilisant ce code ? Raison : deux éléments distincts de $\varphi(\mathbb{F}_2^6)$ ont toujours au moins $16 > 7 + 7$ coordonnées différentes. Étant donné un élément r quelconque de $\mathbb{F}_2^{32}$, il existe donc au plus un unique élément de Image(φ) qui diffère en au plus 7 coordonnées de r.

Preuve : Il faut montrer que $c = a - b = a + b \neq 0^{32}$ a au moins 16 coordonnées non nulles si a et b sont deux éléments distincts de $\varphi(\mathbb{F}_2^6)$. Si v_6 intervient dans $c = (c_1, \dots, c_{32})$, alors $c_{2i-1} \neq c_{2i}$ pour $i = 1, \dots, 16$ et c a donc exactement 16 coordonnées non nulles. Si v_6 n'intervient pas on raisonne similairement sur v_5 etc.

À partir de 8 erreurs, il n'y a plus forcément un candidat évident pour $m = r + e$: Exemple $r = 0^8 1^{24}$ s'obtient soit en ajoutant 8 erreurs $m = v_1 = r + (1^8 0^{24})$ soit $v_2 = r + (0^8 1^8 0^{16})$. On peut donc essayer de deviner m mais on se trompera souvent. Par contre, on peut toujours détecter l'occurence de $\leq$ 15 erreurs (mais on ne pourra corriger correctement qu'en présence d'au plus 7 erreurs) car il faut changer au moins 16 coordonnées pour transformer un élément du code $\varphi(\mathbb{F}_2^6)$ en un autre élément de $\varphi(\mathbb{F}_2^6)$ qu'on pourra confondre avec l'élément émis.

Vocabulaire

On note $\mathbb{F}_2$ le corps à deux éléments dont on désignera les éléments par 0 et 1. $\mathbb{F}_2^n$ désigne l'espace vectoriel de dimension n sur $\mathbb{F}_2$. Dans le monde des codes, on notera souvent des vecteurs comme des mots obtenus

en concaténant les coordonnées.

Attention : Ne confondez pas mots de codes et l'écriture binaire d'entiers naturels : L'addition de deux mots de codes se fait coordonné par coordonné. (Il n'y a pas de retenue, elle n'a pas de sens dans ce contexte !)

Norme de Hamming sur $\mathbb{F}_2^n$: nombre de coefficients 1 d'un élément. Distance de Hamming entre deux éléments a et b dans $\mathbb{F}_2^n$: norme de Hamming de la somme (ou différence) $a + b$. La distance de Hamming de a à b est le nombre de coefficients de a qu'il faut changer pour obtenir b. C'est une distance : valeurs dans les réels positifs ou nuls, nul si et seulement si $a = b$, symétrique et vérifiant l'inégalité du triangle.

Code correcteur linéaire par blocs sur $\mathbb{F}_2$ (ou simplement code correcteur dans la suite) : Sous-espace vectoriel de $\mathbb{F}_2^n$. Les paramètres d'un tel code C sont (n, k, δ) si C est un sous-espace vectoriel de dimension k dans $\mathbb{F}_2^n$ et si δ est la norme de Hamming minimal parmi les éléments non nuls de C. Le choix d'une base $v_1, \ldots, v_k$ de C fournit une application linéaire bijective $\varphi : \mathbb{F}_2^k \longrightarrow C$. Souvent on se donne un code sous la forme d'une injection linéaire $\varphi : \mathbb{F}_2^k \longrightarrow \mathbb{F}_2^n$ qui prend alors la forme d'une matrice $n \times k$ coefficients dans $\{0, 1\}$ (représentant les deux éléments du corps $\mathbb{F}_2$ deux éléments).

$n - k$ est la redondance et $n/k \geq 1$ le coefficient de redondance du code.

On utilise un tel code pour transmettre un élément m de C (ou de $\mathbb{F}_2^k$ en le composant d'abord avec φ) à travers un canal bruité qui ajoute une erreur e (un élément quelconque de $\mathbb{F}_2^n$) au message m. On reçoit donc $r = m + e$ et on veut récupérer m si possible. On dit que C corrige t erreurs si c'est toujours possible pour e de norme de Hamming au plus t (c'est- dire si au plus t coordonnées de m ont été altérés durant la transmission). On dit que C détecte s erreurs si on s'apercevoit de la présence d'au plus s erreurs lors de la réception.

Théorème 2.1. *Un code de paramètres (n, k, δ) détecte toujours $\delta - 1$ erreurs et corrige $\lfloor (\delta - 1)/2 \rfloor$ erreurs.*

Preuve. Comme deux éléments de C diffèrent en au moins δ coordonnées, on ne peut pas changer un élément de C en un autre élément de $\mathcal{C}$ en modifiant moins de δ coordonnées. En recevant un élément r de C, on a donc $r = m$ (où m est l'élément de C qui a été envoyé), à moins que $r = m + e$ avec e contenant au moins δ coordonnées égales 1. Ceci montre que le code C permet de détecter au moins $\delta - 1$ erreurs.

Si le mot r reçu est distance au plus $\lfloor (\delta - 1)/2 \rfloor) < \delta/2$ d'un mot m de C, ce mot m est unique. En effet, si deux mots distincts m et m' de C sont distance $< \delta/2$ de r, alors la distance de m à m' est strictement plus petite que δ par l'inégalité du triangle. (Rappel : distance=distance de Hamming=nombre de coefficients différents.) □

Théorème 2.2. *Les paramètres (n, k, δ) d'un code vérifient la borne de Singleton :*

$$n - k \geq \delta - 1 \ .$$

Preuve. Sinon il existe un code C de paramètres (n, k, δ) avec $\delta \geq n - k + 2$. Comme C est de dimension k, il existe un ensemble de k indices tel que la projection sur ces k indices est une bijection de C vers $\mathbb{F}_2^k$. Supposons pour simplifier que ces k indices correspondent aux k coordonnées initiales. Il existe alors une base de C formée par les k lignes d'une matrice $k \times n$ qui commence par une matrice identité suivie de $n-k$ colonnes additionelles. Un tel vecteur de base a au plus $1+n-k$ coordonnées non nulles en contradiction avec $\delta \geq n - k + 2$.

Code systématique : $\varphi(m)$ est de la forme $(m, \psi(m))$ (à permutations des coordonnées près). Le message d'origine non codé m de $\mathbb{F}_2^k$ se retrouve donc 'en clair' parmi les coefficients du mot codé $\varphi(m)$ de $\mathbb{F}_2^n$.

Matrice de contrôle : matrice d'une application linéaire de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^{n-k}$ qui a comme noyau le code correcteur (de paramètres (n, k, δ)). Un vecteur colonne r de $\mathbb{F}_2^n$ appartient au code C si et seulement si Hr est le vecteur identiquement nul pour H une matrice de contrle (qui n'est pas unique).

En recevant un message r codé par un code C, on applique une matrice de contrôle (pour le code C) à r. Si le résultat est nul, on suppose que r coincide avec le message émis. Si le résultat est non nul, on corrige r en le remplaçant par un plus proche élément de C (unique s'il y a eu strictement moins que $\delta/2$ erreurs durant la transmission). On retrouve ainsi le message émis s'il y a eu strictement moins de $\delta/2$ erreurs durant la transmission.

2.2 Exemples triviaux : Codes répétition

Le message m est un élément du corps $\mathbb{F}_2 = \mathbb{F}_2^1$ (un seul bit). On le répète n fois. Paramètres $(n, 1, n)$. Pas très intéressant en pratique.

2.3 Exemples faciles : codes de parité

En ajoutant au message $m = (m_1, \ldots, m_k)$ dans $\mathbb{F}_2^k$ un bit de parité $p = m_1 + \ldots + m_k$, on obtient un code de paramètres $(k+1, k, 2)$. Il permet donc de détecter une erreur (mais pas de la corriger).

En ayant des messages dans $\mathbb{F}_2^{kl}$ (avec $k, l > 1$) on peut former une matrice (tableau) $k \times l$ et ajouter un bit de parité par ligne et par colonne. On obtient ainsi un code de paramètres $(kl + k + l, kl, 3)$ qui détecte deux erreurs et corrige une erreur.

(Un ou deux exemples $2 \times 2, 2 \times 3$ au tableau.)

2.4 Codes pairs et impairs

L'application $\mathbb{F}_2^n \ni (x_1, \ldots, x_n) \longmapsto \sum_i x_i \pmod 2$ est un morphisme de groupe de $\mathbb{F}_2^n$ dans (le groupe additif) $\mathbb{F}_2$ qui correspond la longueur de Hamming modulo 2.

Un code C est pair s'il est dans le noyau de ce morphisme. Un code C est impair sinon. Tout code C de $\mathbb{F}_2^n$ se prolonge en un code pair de $\mathbb{F}_2^{n+1}$ en

ajoutant un dernier bit de parité (voir aussi ci-dessus). Cette construction est sans intérêt pour un code pair mais elle transforme un code impair de paramètres (n, k, δ) en un code pair de paramètres $(n+1, k, \delta+1)$ si δ est impair. Elle augmente donc la distance minimale de 1 dans le cas d'un code avec distance minimale impaire (un tel code est forcément impair). Cela ne permettra pas de corriger plus d'erreurs mais de detecter une erreur de plus (toujours dans le cas d'un code impair avec distance minimale impaire).

Code parfait : Un code $(n, k, \delta = 2t+1)$ tel que tout élément de $\mathbb{F}_2^n$ est distance de Hamming au plus t d'un élément de $C = \varphi(\mathbb{F}_2^k)$.

Un code de paramètres $(n, k, 2t+1)$ est parfait si et seulement si

$$\sum_{j=0}^{t} \binom{n}{j} = 2^{n-k} \ .$$

(Preuve : La boule fermée de rayon (de Hamming) t contient $\sum_{j=0}^{t} \binom{n}{j}$ éléments.)

Exemple : Un code de paramètres $(7, 4, 3)$ est parfait : $(1+7) = 8 = 2^3$ et $2^3 \cdot 2^4 = 2^7$. Les boules fermées de rayon 1 centrées sur les éléments de $\mathcal{C}$ pavent donc l'espace $\mathbb{F}_2^7$ tout entier.

On peut donner un code par une matrice génératrice dont les lignes forment une base. Essayons de construire une matrice génératrice pour un code $(7, 4, 3)$. On peut supposer que les 4 premières colonnes sont lin indép. et on complète pour obtenir la solution (essentiellement unique)

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Ce code est bien de minimum 3. (Facile à vérifier.)

Le code pair associé (en ajoutant un bit de parité) est de paramètres $(8, 4, 4)$.

Comment comprendre les codes linéaires. Quitte à permuter les coordonnées, un code linéaire est toujours systémique. On peut alors interpréter "les colonnes ajoutées" comme des "bit de parités" pris sur des sous-ensembles d'indices bien choisies. Cela crée de la redondance dans le message qui permet de détecter et de corriger un certain nombre d'erreurs.

Étant donné un code linéaire (n, k, δ), c'est facile de coder un message, de détecter $< \delta$ erreurs et de décoder un message codé sans erreurs (c'est-à-dire un mot du code).

Corriger les erreurs est plus délicat (cela dépend du code en question).

Un problème central en théorie des codes est l'existence (et la construction) de bons codes qui permettent facilement de corriger les erreurs. Construire de tels codes est un sujet actif impliquant des maths difficiles.

3 Révision

3.1 Indispensable

Pgcd, nombres premiers, factorisation des entiers.
Bases. Algorithmique pour l'arithmétique (les opérations usuelles).
Algorithme d'Euclide étendu, Bézout.
Exponentiation rapide
Test de Fermat (c'est un "test", pas une preuve).
Théorème des restes chinois
Notions d'anneau et de corps : les exemples $\mathbb{Z}/N\mathbb{Z}$.
? ? ? (Tout ce que j'ai oublié de mentionner et qui est indispensable)

3.2 Important

Cryptographie, chiffrement, déchiffrement, clé publique, clé secrète.
Chiffrement de César, par substitution (permutation des lettres), attaque par analyse des fréquences.
Changements de base. Problèmes liés aux grands entiers.
Notion de groupe (abélien), notation additive, multiplicative, table d'un groupe fini.
Indicatrice d'Euler, (groupe des) éléments inversibles dans $\mathbb{Z}/N\mathbb{Z}$.
Ordre d'un élément dans un groupe, théorème de Lagrange (l'ordre d'un élément dans un groupe fini divise le cardinal du groupe.)
Comprendre RSA et savoir l'appliquer.
Structure d'une application d'un ensemble fini dans lui-mme. Savoir quoi ressemble le graphe d'une application d'un ensemble fini dans lui-meme. (Nombre fini de cycles avec des arbres attaches.)
Définition et paramètres d'un code correcteur linéaire sur le corps $\mathbb{F}_2$. Distance de Hamming. Lien avec l'algèbre linéaire vue en L1.
Savoir calculer/estimer quelques complexités simples (et avoir compris la différence entre complexité polynomiale et exponentielle).
? ? ? (Tout ce que j'ai oublié de mentionner et qui est important.)

3.3 Notions plus secondaires et plus avancées

Polynmes et bases : similarités, différences.
Algorithme de Karatsuba.
Test de Rabin-Miller.
Différence entre test et preuve (c'est un peu plus que secondaire).
Comprendre RSA " fond".
Théorème des anniversaires.
L'algo de recherche d'un cycle dans un graphe orienté (lièvre et tortue).
Comprendre ρ de Pollard.
Générateurs de $(\mathbb{Z}/p\mathbb{Z})^*$ pour p premier.

4 Exos supplémentaires

4.1 Méthode ρ de Pollard

Exercice 1. Dessiner le graphe orienté dont les sommets sont éléments de $\mathbb{F}_{13} = \mathbb{Z}/13\mathbb{Z}$ avec $a \longmapsto a^2 + 3 \pmod{11}$ comme artes orientées.

Factoriser 143 en utilisant l'algorithme ρ de Pollard avec l'application $x \longmapsto x^2 + 3$ en partant de $x_0 = 0, 1, 2$.

Y a-t-il des x_0 qui marchent mieux que d'autres ?

Exercice 2. Dessiner les graphes pour toutes les applications possibles de la forme $x \longmapsto x^2 + a \pmod{5}$.

Mme chose pour $x \longmapsto x^2 + a \pmod{7}$.

Déterminer le nombre d'itérations de l'algorithme ρ (appliqué un entier composé N contenant un facteur 5 ou 7) avant de tomber sur un multiple de 5, sur un multiple de 7 pour tous les/quelques choix de $x \longmapsto x^2 + a$ et pour tous les/quelques x_0 possibles.

Exercice 3. Étudier quelques applications affines sur les anneaux $\mathbb{Z}/N\mathbb{Z}$, par exemple $x \longmapsto 3x + 2 \pmod{7}$ et $x \longmapsto 2x + 1 \pmod{11}$ en calculant le graphe orienté associé.

Que constatez-vous ?

4.2 Codes correcteurs d'erreurs

Exercice 1. On regarde le code C sur $\mathbb{F}_2$ donné par

$$(a, b, c, d) \longmapsto (a, b, c, d, a + b, c + d, a + c, b + d) \ .$$

Quelle est sa dimension ? Écrire tous ses éléments.

Déterminer ses paramètres (n, k, δ) (o $n > k$ est la dimension de l'espace ambiant, k la dimension de C et δ la norme minimal de Hamming des éléments non nuls de C).

Combien d'erreurs C peut-il détecter, corriger ?

Un mot transmis par C est arrivé sous la forme

$$(0, 1, 0, 0, 1, 1, 0, 0) \ .$$

Que peut-on dire ?

On regarde maintenant le code

$$(a, b, c, d) \longmapsto (a, b, c, d, a + b, c + d, a + c, b + d, a + b + c + d) \ .$$

Dimension, paramètres ?

Construire un code de paramètres $(7, 4, 3)$ et l'augmenter en un code de paramètres $(8, 4, 4)$ en ajoutant un dernier bit de parité.

Comparer ces deux derniers codes au deux codes du début de l'exercice.