«Qui est-ce ?» version Alice et Bob

par Justin Vast

Un rappel

Soit q une puissance d'un nombre premier, et soit $\mathbb{F}_q$ l'unique corps (à isomorphisme près) à q éléments.

Un **code** de **longueur** n sur l'**alphabet** $\mathbb{F}_q$ est un sous-ensemble non-vide $\mathcal{C} \subseteq \mathbb{F}_q^n$, et ses éléments sont les **mots du code**.

Un **code linéaire** $\mathcal{C}$ de longueur n est un sous-espace vectoriel de $\mathbb{F}_q^n$.

La **distance de Hamming** entre deux mots $x, y \in \mathbb{F}_q^n$ est le nombres de coordonnées distinctes entre x et y. Plus formellement, $d(x,y) := \#\{i \in \{1, \cdots, n\} | \, x_i \neq y_i\}$.

La **distance minimale** d'un code $\mathcal{C} \subseteq \mathbb{F}_q^n$ est définie par

$$d(\mathcal{C}) := \min \{d(x,y) \,|\, x, y \in \mathcal{C}, x \neq y\}$$

Un code **détecte** $d(\mathcal{C}) - 1$ erreurs, et en **corrige** $\left\lfloor \frac{d(\mathcal{C})-1}{2} \right\rfloor$.

Si $\mathcal{C} \subseteq \mathbb{F}_q^n$ est un code linéaire de dimension k :

On dit que $M \in \mathcal{M}(n \times k, \mathbb{F}_q)$ est une **matrice génératrice** du code $\mathcal{C}$ si ses colonnes forment une base de $\mathcal{C}$ en tant que $\mathbb{F}_q$-espace vectoriel.

On dit que $A \in \mathcal{M}((n-k) \times n, \mathbb{F}_q)$ est une **matrice de contrôle** du code $\mathcal{C}$ si $\mathcal{C} = \operatorname{Ker} A$.

Proposition : La distance minimale d'un code linéaire $\mathcal{C}$ est la plus petite quantité de colonnes de A nécessaires pour former un ensemble linéairement dépendant.

Pour $v \in \mathbb{F}_q^n$, le vecteur $s(v) := Av \in \mathbb{F}_q^{n-k}$ est appelé **syndrome** de v, où A est une matrice de contrôle de $\mathcal{C}$. Si $z = x + e$ (avec $x \in \mathcal{C}$), alors $s(z) = s(e)$, donc le syndrome du message reçu est le syndrome des erreurs commises.

Le **code de Hamming** $\mathcal{C} \subseteq \mathbb{F}_2^7$ est le code dont une matrice de contrôle et une matrice génératrice sont respectivement

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ et } M = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Sa distance minimale vaut 3 (il corrige donc une erreur).

Par la structure du code de Hamming, pour corriger un mot $v \in \mathbb{F}_2^7$ avec au plus une erreur, on regarde le syndrôme de v : $s(v) = A \cdot v$; s'il est nul c'est qu'il n'y a pas d'erreur ; sinon, $s(v)$ est la i^e colonne de A (pour un certain i), et la correction de v est $v_{\text{corr}} := v + e_i$, où $e_i \in \mathbb{F}_2^7$ est nul sauf en sa i^e composante.

Jeu 0 : Alice et Bob décident de faire un jeu semblable à *Qui est-ce ?* avec des nombres. Alice pense à un nombre entre 0 et 15 (inclus), et Bob doit essayer de déterminer ce nombre à l'aide de questions auxquelles Alice ne peut répondre que par OUI ou par NON.

Bob étant adepte des énigmes, il se demande quelle sera la quantité *nécessaire et suffisante* de questions qu'il devra poser à Alice, afin d'être sûr de connaître le nombre auquel Alice pense.

Une stratégie naïve

Stratégie 0 : Poser au plus 16 questions :

→ Q0 : Est-ce que le nombre auquel tu penses est 0 ?

→ Q1 : Est-ce que le nombre auquel tu penses est 1 ?

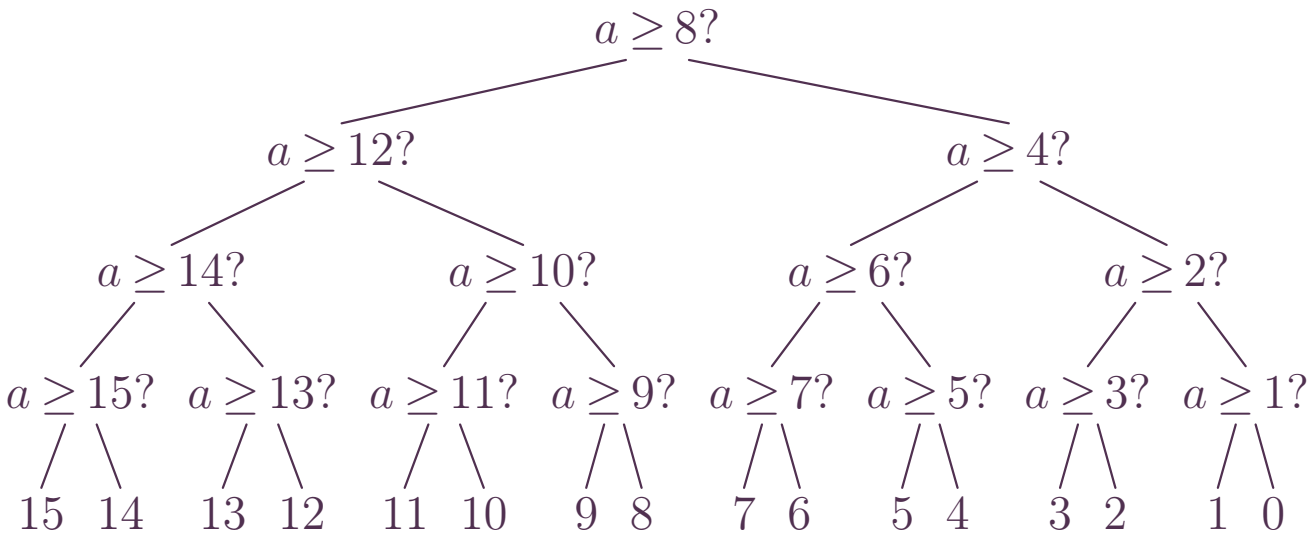
→ Q2 : Est-ce que le nombre auquel tu penses est 2 ?

...

→ Q15 : Est-ce que le nombre auquel tu penses est 15 ?

Cette stratégie fonctionne, mais elle est loin d'être optimale... Néanmoins, nous savons maintenant que 16 est une quantité suffisante de questions à poser.

Stratégie 1 : Notons a le nombre auquel Alice pense. On applique l'algorithme de recherche dans un **arbre binaire de recherche** (ABR) :



On en déduit que 4 est une est une quantité *suffisante* de questions à poser, et clairement, cette quantité est également *nécessaire*.

Remarquons que si $1 := \text{OUI}$ et $0 := \text{NON}$, alors la suite des réponses d'Alice est l'écriture de a en base 2. Par exemple, si $a = 7$, la suite des réponses d'Alice est 0111, et $7 = 0111_2$.

Alice et Bob se sont bien amusés, mais c'était un peu facile, soyons honnêtes... il est maintenant temps de compliquer les choses !

Une variante du jeu précédent

Jeu 1 : Alice pense de nouveau à un nombre entre 0 et 15 (inclus), et Bob doit toujours essayer de déterminer ce nombre à l'aide de questions auxquelles Alice ne peut répondre que par OUI ou par NON. *Cependant*, Alice peut faire exprès de donner *au plus* une fausse réponse (répondre OUI au lieu de NON, et vice versa).

Cette fois-ci, quelle sera la quantité *nécessaire et suffisante* de questions que Bob devra poser à Alice, afin d'être sûr de connaître le nombre auquel Alice pense ?

Une plutôt bonne stratégie

Stratégie 0 : Alice pense à $a = 7$. Bob a imaginé un algorithme basé sur la stratégie de l'ABR du jeu précédent ; voici des questions-réponses possibles :

$a \geq 8$?	NON	$a \geq 8$?	NON	$a \geq 8$?	NON	$a \geq 8$?	OUI
$a \geq 8$?	NON	$a \geq 8$?	NON	$a \geq 8$?	NON	$a \geq 8$?	NON
$a \geq 4$?	NON	$a \geq 4$?	OUI	$a \geq 4$?	OUI	$a \geq 8$?	NON
$a \geq 4$?	OUI	$a \geq 4$?	OUI	$a \geq 4$?	OUI	$a \geq 4$?	OUI
$a \geq 4$?	OUI	$a \geq 6$?	OUI	$a \geq 6$?	OUI	$a \geq 6$?	OUI
$a \geq 6$?	OUI	$a \geq 6$?	OUI	$a \geq 6$?	OUI	$a \geq 7$?	OUI
$a \geq 7$?	OUI	$a \geq 7$?	OUI	$a \geq 7$?	OUI		
		$a \geq 7$?	NON				
		$a \geq 7$?	OUI				

Voyez-vous l'algo qui se cache derrière ?

L'algorithme

$A \longleftarrow 0$

b $\longleftarrow$ True (**b** = « aucune erreur n'a encore été commise par Alice »)

FOR $i = 0 \to 3$:

  « $a \geq A + 2^{3-i}$?» : $a_i \longleftarrow$ réponse

  IF **b** :

    « $a \geq A + 2^{3-i}$?» : $b_i \longleftarrow$ réponse

    IF $a_i \neq b_i$:

      b $\longleftarrow$ False

      « $a \geq A + 2^{3-i}$?» : $c_i \longleftarrow$ réponse

      $A \longleftarrow A + 2^{3-i} c_i$

    ELSE : $A \longleftarrow A + 2^{3-i} a_i$

  ELSE : $A \longleftarrow A + 2^{3-i} a_i$

RETURN A

Améliorons les choses…

On peut montrer que grâce à l'algo qui vient d'être présenté, il est *suffisant* de poser 9 questions ; mais est-ce *nécessaire* d'en poser autant ?

Il serait judicieux d'utiliser la théorie des codes correcteurs d'erreurs !

Notons $n \in \mathbb{N}$ la quantité *nécessaire* et *suffisante* de questions que Bob doit poser à Alice pour s'assurer d'avoir deviné $a \in \{0, 1, \cdots, 15\}$. Par la *stratégie 0*, $n \leq 9$.

Attention, Bob peut potentiellement avoir deviné a en utilisant moins de n questions ! Quitte à poser des questions supplémentaires inutiles, on peut supposer que Bob posera toujours n questions.

Les réponses d'Alice seront représentées sous la forme de vecteurs $\in \mathbb{F}_2^n$, où la composante i du vecteur représente la réponse à la question i de Bob (rappel : $1 := \text{OUI}$ et $0 := \text{NON}$).

On suppose que Bob pose les bonnes questions, i.e. il pourra toujours déterminer a après avoir posé ses n questions. Note importante : Bob s'est fixé une première question à poser, mais les questions suivantes dépendent potentiellement des réponses déjà données par Alice.

Puisque Bob va poser n questions, il y a 2^n possibilités de réponses à celles-ci.

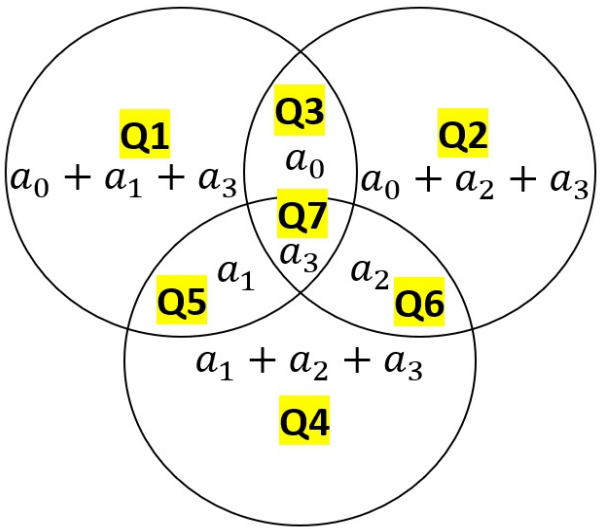
On sait qu'Alice peut commettre volontairement au plus 1 erreur parmi ses n réponses ; donc pour chaque $a \in \{0, 1, \cdots, 15\}$, il existe exactement $n+1$ possibilités de réponses de la part d'Alice.

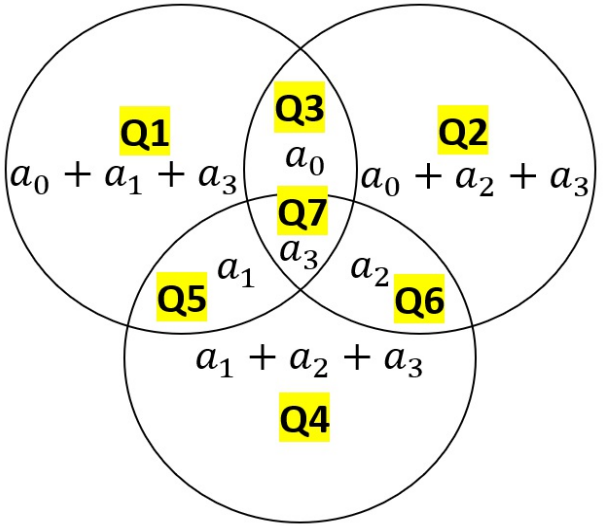
Par hypothèse, Bob est capable de déterminer (sans aucun doute possible) le nombre auquel Alice pense, en utilisant uniquement les réponses données par Alice. Ainsi, $16\,(n+1) \leq 2^n$, d'où $n \geq 7$.

7 est donc une quantité nécessaire de questions à poser ; mais est-elle suffisante ? Nous allons voir que oui !

Stratégie 1 : Demander à Alice de précalculer l'écriture binaire du nombre a auquel elle pense (e.g. 0010_2 pour 2, 1011_2 pour 11) (cette demande n'est pas considérée comme une question). Disons que l'écriture binaire de a est $a_{\text{bin}} = a_0a_1a_2a_3 \in \mathbb{F}_2^4$. Les 7 questions sont les suivantes (l'ordre n'a pas d'importance en réalité) :

Q1 : $a_0 + a_1 + a_3 = 1 (\in \mathbb{F}_2)$?	Q5 : $a_1 = 1$?
Q2 : $a_0 + a_2 + a_3 = 1$?	Q6 : $a_2 = 1$?
Q3 : $a_0 = 1$?	Q7 : $a_3 = 1$?
Q4 : $a_1 + a_2 + a_3 = 1$?	





Remarquons que poser ces questions revient à demander de calculer $M \cdot a_{\text{bin}}$, où

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

et M est exactement une matrice génératrice du *code de Hamming*, qui rappelons-le, corrige 1 erreur, ce qui est exactement ce que nous recherchions !

Il faut maintenant corriger l'erreur, en utilisant le syndrôme du vecteur v des réponses aux questions (on a $v := M \cdot a_{\text{bin}} + e$ où e est un certain vecteur d'erreurs) :

$$s(v) = A \cdot v \in \mathbb{F}_2^3$$

où

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

Si $s(v) = 0 \in \mathbb{F}_2^3$, c'est qu'Alice n'a commis aucune erreur, et la correction de v est simplement $v_{\text{corr}} := v$.

Sinon, $s(v)$ est la i^e colonne de A pour un certain i, et la correction de v est $v_{\text{corr}} := v + e_i$ où $e_i \in \mathbb{F}_2^7$ est le vecteur qui vaut 0 partout sauf en la i^e composante.

Maintenant, nous savons que $v_{\text{corr}} = M \cdot a_{\text{bin}}$; ainsi, si $P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$, on a

$$P \cdot v_{\text{corr}} = P \cdot M \cdot a_{\text{bin}} = \mathbb{I}_4 \cdot a_{\text{bin}} = a_{\text{bin}}$$

(rappelons que $M = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$), et on récupère finalement a à partir de a_{bin} !

En conclusion, 7 est bien la quantité *nécessaire* et *suffisante* de questions que Bob doit poser à Alice afin de déterminer le nombre auquel elle pense.

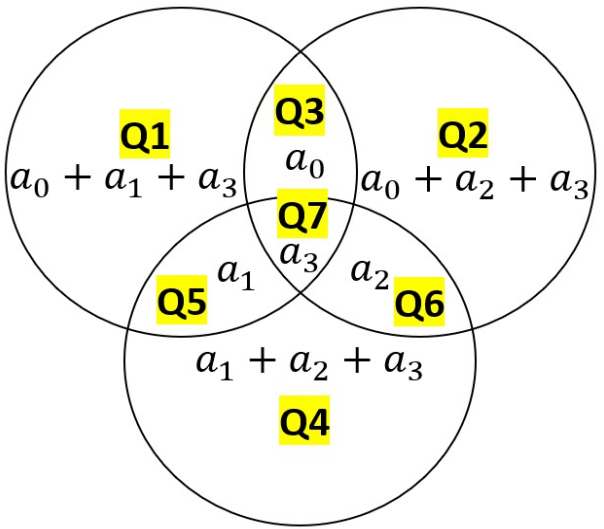
Voyons un exemple… Alice pense à $a = 12$.

Bob lui demande d'écrire a en base 2 : $a_{\text{bin}} = a_0 a_1 a_2 a_3 = 1100$

Bob pose les 7 questions (i.e. il lui demande de calculer $M \cdot a_{\text{bin}}$), et voici le vecteur des réponses d'Alice : $v = 0011100$.

Bob calcule le syndrome de v : $s(v) = A \cdot v = 010$, ce qui est exactement la deuxième colonne de A. Par conséquent, $v_{\text{corr}} = v + e_2 = 0111100$.

Enfin, Bob calcule $P \cdot v_{\text{corr}} = 1100$, c'est-à-dire l'écriture de $a = 12$ en base 2.



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

$$M = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Un jeu alternatif un peu plus général

Remarquons que pour cette dernière stratégie, les questions qui ont été posées ne dépendent pas des réponses données par Alice.

Analysons une version alternative du *jeu 1* :

Jeu 2 : Alice pense à un nombre (naturel) a compris entre 0 et $m-1$ (inclus), et Bob doit essayer de déterminer ce nombre à l'aide de questions auxquelles Alice ne peut répondre que par OUI ou par NON. Alice peut commettre volontairement au plus r erreurs. On suppose que Bob a préfixé ses questions, càd que celles-ci ne dépendront pas des réponses données par Alice.

Quelles est la quantité n nécessaire et suffisante de questions que Bob doit poser à Alice afin de déterminer a avec exactitude ?

La question du *jeu 2* est un problème difficile à résoudre dans le cas général. Cependant :

Théorème. *Les assertions suivantes sont équivalentes :*

i. *Il faut et il suffit à Bob de poser* n *questions à Alice pour déterminer* a *selon les règles du* jeu 2.

ii. *Il existe un code* $\mathcal{C} \subseteq \mathbb{F}_2^n$ *(pas forcément linéaire), de taille* $|\mathcal{C}| \geq m$*, et de distance minimale* $d(\mathcal{C}) \geq 2r+1$ *; et pour tout code* $\mathcal{C}' \subseteq \mathbb{F}_2^{n'}$ *avec* $n' < n$*, si* $|\mathcal{C}'| \geq m$*, alors* $d(\mathcal{C}') < 2r+1$.

La preuve n'est pas excessivement compliquée.

Notons

$$b_q(n,r) := \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$$

Borne de Hamming. *Soit $\mathcal{C} \subseteq \mathbb{F}_q^n$ un code (pas forcément linéaire) de distance minimale $d(\mathcal{C}) \geq 2r+1$. Alors*

$$|\mathcal{C}| \cdot b_q(n,r) \leq q^n$$

Indication de preuve : $\bigsqcup_{x \in \mathcal{C}} B[x,r] \subseteq \mathbb{F}_q^n$, où $B[x,r]$ est la boule fermée de centre x et de rayon r (pour la distance de Hamming).

Avec les données du *jeu 1*, on a $|\mathcal{C}| = 16$, $q = 2$, $r = 1$, la borne de Hamming nous donne :

$$16(n+1) \leq 2^n, \text{ càd } n \geq 7$$

Voici d'autres bornes spécifiques aux codes linéaires $\mathcal{C} \subseteq \mathbb{F}_q^n$ avec $|\mathcal{C}| = q^k$

Borne de Singleton. $k + d(\mathcal{C}) \leq n + 1$

Borne de Plotkin. $d(\mathcal{C}) \leq \dfrac{n q^{k-1}(q-1)}{q^k - 1}$

On a également une borne « dans l'autre sens », qui donne cette fois-ci l'existence de codes (linéaires) :

Borne de Gilbert-Varshamov. *Si*

$$q^{n-k+1} > b_q(n, 2r)$$

alors il existe un code linéaire $\mathcal{C} \subseteq \mathbb{F}_q^n$ *avec* $|\mathcal{C}| = q^k$ *et de distance minimale* $d(\mathcal{C}) \geq 2r+1$.

On veut trouver le n minimal pour lequel la borne de Gilbert-Varshamov est satisfaite avec les données du *jeu 1* ($k=4$, $q=2$, $r=1$), càd $2^{n-3} > 1 + n + \frac{n(n-1)}{2}$, ou encore $2^{n-2} > n^2 + n + 2$; on trouve $n=9$ (moins bien que 7).