# Séance 7: Code de Gray

Université Paris-Diderot

Le code de Gray est un type de codage binaire permettant de ne modifier qu'un seul bit à la fois quand un nombre est augmenté d'une unité. On souhaite programmer une fonction qui prend un nombre naturel en notation positionnelle en base 2, appelé ici un code binaire, et qui renvoie le code de Gray correspondant. Ensuite on s'intéressera à la fonction de décodage correspondante.

Voici les algorithmes :

Codage (binaire $\Rightarrow$ Gray) : Si $b_0 b_1 \dots b_k$ est un code binaire, $b_0$ étant le bit de poids fort, alors le code de Gray correspondant est la séquence $g_0 g_1 \dots g_k$, où $g_0 = b_0$ et, pour $1 \leq i \leq k$, $g_i = b_i \oplus b_{i-1}$ ( $\oplus$ désigne le ou exclusif). Par exemple, à partir de $0110$ on obtient $0101$.

Décodage (Gray $\Rightarrow$ binaire) : Si $g_0 g_1 \dots g_k$ est un code de Gray, le code binaire correspondant est $b_0 b_1 \dots b_k$ où pour $0 \leq i \leq k$ $b_i = g_0 \oplus g_1 \oplus \dots \oplus g_i$ (et donc, en particulier, $b_0 = g_0$). Par exemple, à partir de $0101$ on obtient $0110$.

Remarque:
Vous rédigerez les réponses aux exercices dans le fichier fourni `Exos.java`. Il est important de **tester** au fur et à mesure votre code, à l'aide des tests fournis dans le `main`.

**Exercice 1 (codage, $\star$)**
*Écrivez une fonction* `codeGray` *qui prend en entrée un tableau de booléens répresentant un code binaire (*`true` *pour 1 et* `false` *pour 0) et renvoie le tableau de booléens répresentant le code de Gray corréspondant (le bit de poids fort étant celui d'indice 0).* □

**Exercice 2 (décodage, $\star\star$)**
*Écrivez une fonction* `decodeGray` *qui prend en entrée un tableau de booléens répresentant un code de Gray et renvoie le tableau de booléens répresentant le code binaire corréspondant.* □

**Exercice 3 (auxiliaire, $\star$)**
*Écrivez une fonction* `aux` *qui prend en entrée un entier non négatif $n$ et et un entier $m$ (on peut supposer $2^m > n$) et qui renvoie un tableau de booléens de taille $m$ contenant le code binaire de $n$. Par exemple* `aux(3,4)` *renvoie le tableau* `false,false,true,true`. □

**Exercice 4 (Voisinage de Gray, $\star\star$)**
*Écrivez une fonction* `nearGray` *qui prend en entrée deux entiers non négatifs $m$ et $n$, calcule le plus petit $k$ tel que $m, n < 2^k$, et renvoie* `true` *si les codes de Gray de longuer $k$ de $m$ et de $n$ sont voisins, et* `false` *sinon. Deux codes de Gray sont voisin s'ils diffèrent sur un indice au plus. Par exemple,* `nearGray(6,9)` *renvoie* `true` *car les codes de Gray sur 4 bits de 6 et 9 sont* `false,true,false,true` *et* `true,true,false,true`*, et ces deux tableaux différent sur un seul indice. Par contre,* `nearGray(3,5)` *renvoie* `false` *car les codes de Gray de 3 et 5 sur 3 bits sont* `false,true,false` *et* `true,true,true` *respectivement* □