IT_122 – 2021-2022
TP numéro 1 : Polynômes et Reed-Solomon (modulo un nombre premier).

Rappels théoriques

On rappelle qu'un polynôme de degré n à coefficients dans un certain corps K (qui ici sera ici les entiers modulo un premier) est une fonction $K \to K$ de la forme

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = \sum_{k=0}^{n} a_k x^k,$$

où $a_k \in K$ et $a_n \neq 0$. On a vu au cours que l'on peut additionner, soustraire, multiplier des polynômes, ainsi qu'effectuer des divisions euclidiennes (avec reste). On a également vu que si on se donne $n+1$ points du plan K^2 $(x_0, y_0), \ldots, (x_n, y_n)$, avec tous les x_n distincts 2 à 2, alors il y a un *unique* polynôme $L(x)$ de degré $\leq n$ qui passe par ces points, autrement dit pour lequel $L(x_k) = y_k$ pour $k = 0, \ldots, n$:

$$\ell_i(x) = \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k}, \qquad i = 0, 1, \ldots, n$$

$$L(x) = \sum_{i=0}^{n} y_i \cdot \ell_i(x)$$

(Bon, on doit évidemment supposer que n est plus petit que le nombre d'éléments dans le corps K, si par exemple on prend le corps à 2 éléments $\{0, 1\}$, on ne peut pas trouver 4 points dans K^2 avec premières coordonnées distinctes.)
On rappelle également le principe des codes de Reed-Solomon :
– On prend une liste de nombres, par exemple $[3, 2, 5, 7, 2]$, de longueur 5.
– On calcule le polynôme de Lagrange $L(x)$ qui passe par $(0, 3), (1, 2), (2, 5), (3, 7), (4, 2)$, donc 5 points.
– On ajoute $2n$ points à cette liste, qui permettront de corriger n éventuelles erreurs dans la transmission. Par exemple, si $n = 2$, on calcule $L(5)$, $L(6)$,$L(7)$,$L(8)$ et on les ajoute à la liste. On transmet alors cette liste.
– Du côté du receveur, une fois la liste reçue (avec des erreurs éventuelles), on sait que la liste originale est de longueur 5 (dans notre cas). On prend les uns après les autres tous les polynômes de Lagrange passant par (dans notre cas, donc) 5 des 9 points reçus. Si un de ces polynômes passe par 7 points (au moins), on considère que c'est le bon polynôme, à savoir $L(x)$. (Ceci est assuré si il y a au maximum 2 erreurs, s'il y en a plus on ne peut a priori pas retrouver la liste de base.)
– Une fois $L(x)$ retrouvé, on recalcule $L(0), L(1), \ldots, L(4)$ pour retrouver les valeurs de la liste originale.

Le but de ce TP est de décoder des listes obtenues avec Reed-Solomon dans lesquelles quelques erreurs se sont glissées, dans les nombres modulo un nombre premier p. Donc, toutes les multiplications et additions sont modulo p, et, par exemple, $\dfrac{x - x_k}{x_i - x_k}$ doit se comprendre comme la multiplication de $(x - x_k)$ par l'inverse modulo p du nombre $(x_i - x_k)$. Les polynômes doivent être entrés comme la liste de leurs coefficients : $3x^2 + 2x - 1$ devient `[-1,2,3]`.

Le choix du langage est libre (mais je n'ai pas de compétences autres qu'en Python).

A faire

1. Coder l'algorithme d'Euclide étendu qui calcule les coefficients de Bezout de deux entiers a, b, c'est-à-dire trouver $u, v \in \mathbb{Z}$ tels que $a \cdot u + b \cdot v = \text{PGCD}(a, b)$. (Ceci sera aussi utile dans l'autre TP de mathématiques.)
2. Grâce aux coefficients de Bezout, coder une fonction qui donne l'inverse multiplicatif d'un entier a dans les entiers modulo un nombre premier p. Exemple : dans les entiers modulo 5, l'inverse multiplicatif de 4 est 4 (car $4 \cdot 4 = 16 \equiv 1 \mod 5$).
3. Coder les fonctions d'addition et multiplication de polynômes, le tout modulo un premier p.
4. Coder une fonction permettant d'évaluer un polynôme en un point (également modulo p).
5. Coder une fonction qui calcule le polynôme de Lagrange $L(x)$ avec des coefficients (et des valeurs) dans les entiers modulo un premier p.

Pour info, le nombre de lignes de code au total ne devrait pas dépasser 40 (hors commentaires et sauts de ligne pour la lisibilité), en comptant large.

Chaque groupe (3 ou 4 étudiants par groupe) recevra un message codé par Reed-Solomon (dans les entiers modulo un premier qui sera précisé) où des erreurs se sont glissées, et qu'il faudra décoder en utilisant les fonctions codées ci-dessus. Il y aura bien sûr des indications sur le nombre d'erreurs et la longueur de la liste de base. (En fait, chaque groupe recevra un fichier texte avec toutes les infos.) Une fois décodé, il faudra transformer cette liste en chaîne de caractères, chaque membre de la liste étant le numéro unicode d'un caractère.

A rendre pour le jeudi 23 décembre

Dans un fichier compressé appelé `Nom1_Nom2_etc.zip` (ou une autre compression standard plutôt que zip), où Nom1, etc, sont les noms des membres du groupe :
– Votre code **commenté**,
– Le résultat du décodage de Reed-Solomon,
– Deux ou trois paragraphes (pas plus) dans un fichier pdf pour expliquer votre méthode de décodage.