Colle d'informatique n°3.

Compression ternaire

On attachera une grande importance à la concision, à la clarté, et à la précision de la rédaction. On supposera que le langage de programmation utilisé possède deux opérations x `div` y et x `mod` y donnant le quotient et le reste de la division euclidienne de x par y.

Le temps d'exécution $T(f)$ d'une fonction f est le nombre d'opérations élémentaires (addition, soustraction, multiplication, division, affectation, etc.) nécessaire au calcul de f. Lorsque ce temps d'exécution dépend d'un paramètre n, il sera noté $T_n(f)$. On dit que la fonction f s'exécute :

- en temps linéaire en n, s'il existe $K > 0$ tel que pour tout n, $T_n(f) \leqslant Kn$;
- en temps quadratique en n, s'il existe $K > 0$ tel que pour tout n, $T_n(f) \leqslant Kn^2$.

Nombres ternaires

En base 3, les entiers 0, 1, 2, 3, 4, 5, 6, 7, 8 sont représentés par $\underline{00}$, $\underline{01}$, $\underline{02}$, $\underline{10}$, $\underline{11}$, $\underline{12}$, $\underline{20}$, $\underline{21}$, $\underline{22}$. Le chiffre de poids fort de $\underline{bc}$ est `b` ; le chiffre de poids faible est `c`.

Question 1. Écrire la fonction `entier(b, c)` retournant l'entier compris entre 0 et 8 qui s'écrit $\underline{bc}$ en base 3.

Question 2. Soit x un entier vérifiant $0 \leqslant x \leqslant 8$. Écrire une fonction `poidsFort(x)` retournant le chiffre de poids fort de x en base 3. Écrire la fonction `poidsFaible(x)` retournant le chiffre de poids faible de x.

Textes ternaires

Dans ce problème, les textes sont représentés en représentation ternaire. Un savant russe nous a convaincus de la pertinence de ce choix plus compact que la représentation binaire. Un texte est rangé dans un tableau t de N caractères vérifiant $t[i] \in \{0, 1, 2\}$ pour tout i vérifiant $0 \leqslant i < N - 1$; par ailleurs $t[N-1] = X > 2$ (le dernier caractère n'est pas ternaire). On suppose $N \geqslant 1$.

Quelques définitions sont nécessaires : la chaîne de caractères de longueur ℓ démarrant en i est la suite $\langle t[i], t[i+1], \dots, t[i+\ell-1] \rangle$. On dira que deux chaînes $\langle t[i], t[i+1], \dots, t[i+\ell-1] \rangle$ et $\langle t[j], t[j+1], \dots, t[j+\ell'-1] \rangle$ sont égales si $\ell = \ell'$ et $t[i+k] = t[j+k]$ pour $0 \leqslant k < \ell$.

Question 3. Écrire une fonction `longueurMotif`(t, i, j, m) qui retourne, en temps linéaire par rapport à N, la plus grande longueur ℓ d'une chaîne démarrant en i égale à une chaîne démarrant en j. En outre, cette longueur doit vérifier $\ell \leqslant m$.

Question 4. On suppose $i < j$. écrire une fonction `longueurMotifMax`(t, i, j, m) qui retourne, en temps quadratique par rapport à N, la plus grande longueur ℓ d'une chaîne démarrant en $i + k$ égale à une chaîne démarrant en j pour $0 \leqslant k < m$. En outre, on exige $i + k < j$ et $\ell \leqslant m$.

On suppose qu'il existe trois variables globales entières A, L, C.

Question 5. Modifier la fonction précédente pour obtenir la fonction `motifMax`(t, i, j, m) qui rend, en temps quadratique, dans L la plus grande longueur d'une chaîne démarrant en $i+k$ égale à une chaîne démarrant en j pour $0 \leqslant k < m$; qui rend dans A la valeur de k pour lequel $i + k$ est l'indice de départ de cette chaîne de longueur maximale ; qui rend enfin dans C le caractère suivant cette chaîne à partir de j dans t. À nouveau, cette longueur doit vérifier $\ell \leqslant m$. Et on a $i + k < j$ (cf. l'exemple dans la figure suivante).

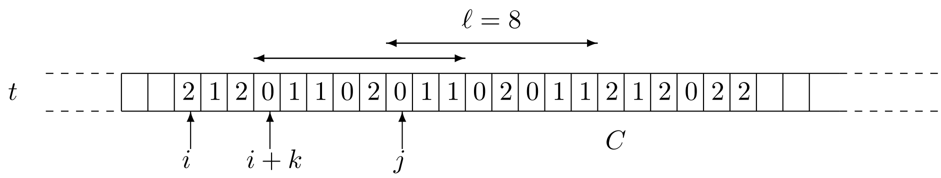

Figure 1: $A = k = 3$; $L = \ell = 8$; $C = 2$

Compression

La méthode de compression de Ziv et Lempel, adoptée dans les commandes `zip` ou `gzip`, consiste à repérer les motifs maximaux déjà rencontrés dans un texte et à indiquer pour chacun d'eux le triplet (A, L, C) calculé dans la question précédente entre toute paire d'indices i et j. Pour mesurer le facteur de compression, nous utilisons le même codage pour ces triplets que pour les caractères du texte, c'est-à-dire le système ternaire dans ce problème.

Question 6. écrire une fonction `imprimerTriplet`(A, L, C) qui imprime les arguments A, L, C sous forme de cinq chiffres consécutifs, les deux caractères ternaires de A, puis les deux caractères ternaires de L, puis le chiffre C, en imposant $0 \leqslant A < 9$, $0 \leqslant L < 9$ et $0 \leqslant C \leqslant 9$.

On suppose à présent que t contient un long texte ternaire commençant par 9 caractères 0 ; en outre, t finit par un caractère `x` spécial $(x > 2)$. On déplace sur ce texte une fenêtre de longueur 18. Au début, cette fenêtre est alignée à gauche sur le début du tableau, et on pose $j = 9$. En régime de croisière, la dixième case de la fenêtre correspond à l'entrée j du tableau t. On recherche, dans la partie gauche de la fenêtre, la chaîne de longueur maximale vérifiant $\ell < 9$ et égale à une chaîne de caractères démarrant en j. On imprime, grâce à la fonction `imprimerTriplet`, le triplet (A, L, C) donné par `motifMax`. Puis, on recentre la fenêtre sur le caractère suivant le caractère d'arrêt C. Ce processus continue jusqu'au bout du tableau t comme indiqué par la figure.

Ainsi pour le texte suivant, on obtient les décompositions de chacune des lignes, soit au total le facteur de compression 30/37 qui serait nettement meilleur dans une base supérieure à 3 et si la taille de la fenêtre était plus grande que 18 (Il y a en effet 30 caractères dans le résultat et 37 dans le texte d'entrée).

t	0 0 0 0 0 0 0 0 0 1 0 2 1 0 2 1 0 1 2 1 0 2 1 0 0 2 1 0 2 1 0 0 2 1 0 0 x
(0,0,1)	0 0 0 0 0 0 0 0 0 1
(0,1,2)	0 0 0 0 0 0 0 0 0 1 0 2
(6,5,1)	0 0 0 0 0 0 1 0 2 1 0 2 1 0 1
(2,6,0)	1 0 2 1 0 2 1 0 1 2 1 0 2 1 0 0
(2,8,1)	0 1 2 1 0 2 1 0 0 2 1 0 2 1 0 0 2 1
(5,2,x)	2 1 0 2 1 0 0 2 1 0 0 x
résultat	0 0 0 0 1 | 0 0 0 1 2 | 2 0 1 2 1 | 0 2 2 0 0 | 0 2 2 2 1 | 1 2 0 2 x

Question 7. Écrire une fonction `compresser`(t) qui imprime sur le terminal de sortie le texte compressé par la méthode précédente.

Pour la décompression, on produit d'abord 9 caractères 0. On considère ensuite tous les triplets (A, L, C) représentés par 5 caractères ternaires consécutifs et on recrée la chaîne originale jusqu'au dernier triplet dont la composante C n'est pas comprise entre 0 et 2.

Question 8. Écrire une fonction `décompresser`(tc) qui prend un texte ternaire `tc` correspondant à du texte compressé et imprime sur le terminal de sortie le texte décompressé correspondant.