TD Maple : d'après X2007 MP/PC

▶ Nous traitons la première partie de l'épreuve d'informatique du concours d'entrée à l'École polytechnique, année 2007, filières MP (option PSI) et PC.

Tableaux en Maple

▶ Une *séquence* en Maple est une suite d'objets séparés par des virgules :

```
s1 := 2,9,3,10 ;
s2 := 3$5 ;
s3 := seq(k^2,k=1..5) ;
```

▶ Un *tableau* à une dimension est défini avec `array` :

```
t1 := array(1..4,[s1]) ;
t3 := array(1..5,[s3]) ;
```

▶ On accède à un élément d'un tableau en donnant le nom du tableau et l'indice de l'élément :

```
t1[2],t3[t1[1]];
```

▶ Une fois construit, un tableau ne peut pas être « agrandi » ou « rétréci » : il faut créer un autre tableau, et copier dedans ce qui nous intéresse. Exemple :

```
tcop := array(1..3) ;
for k from 1 to 3 do
 tcop[k] := t3[k+1]
od;
```

▶ Pour afficher un tableau, utilisez `print`.

Le sujet

▶ Un *texte* est un tableau contenant des entiers compris entre 1 et 255 inclus. La *longueur* de ce texte est le nombre d'éléments du tableau. Une *occurrence* de la valeur v dans un texte t est le nombre d'indices i tels que $t[i] = v$.

Q1 Écrire la fonction `occurrences` qui prend en argument un tableau t et rend un tableau r de taille 255, défini comme suit : pour $k \in [\![1,255]\!]$, $r[k]$ est le nombre d'occurrences de k dans t.

▶ Une *répétition maximale contiguë* d'une valeur v dans un texte t est un couple (i, j) vérifiant les trois conditions suivantes :

- $t[k] = v$ pour tout $k \in [\![i,j]\!]$
- si $i > 1$, alors $t[i-1] \neq v$
- si $j < n$ (où n est la longueur de t), alors $t[j+1] \neq v$.

Nous allons compresser le texte t en remplaçant toute répétition maximale non banale (c'est-à-dire : de longueur au moins 2) par le triplet $(0, j-i, v)$. Par exemple, le texte $(8,8,3,2,3,3,3,3,3,7)$ aura la forme compressée que voici : $(\underbrace{0,1,8}_{8,8}, \underbrace{3}_{3}, \underbrace{1}_{1}, \underbrace{0,5,3}_{3,3,3,3,3}, \underbrace{7}_{7})$.

Q2 Écrire une fonction `repetitionMax` qui prend en argument le tableau t et une position i comprise entre 1 et la longueur de t, et rend la longueur de la répétition maximale qui commence en i. Vous ne regarderez pas les caractères d'indice inférieur à i.

Q3 Écrire une fonction `tailleCodage` qui prend en argument le tableau t et rend la taille n' du texte compressé ; pour l'exemple ci-dessus, vous devez trouver $n' = 12$.

Q4 Écrire une fonction `Codage` qui prend en argument le tableau t et rend la forme compressée t'.

Q5 La fonction de codage que nous venons de définir est-elle injective ? Est-elle surjective ?

Q6 Écrire une fonction `Decodage` qui prend en argument un tableau t' et rend le tableau t dont t' est la forme compressée, lorsqu'un tel tableau t existe. À défaut, votre programme affichera un message indiquant la position à laquelle le décodage a été abandonné.

Q7 A-t-on toujours $|t'| < |t|$? Proposez une amélioration de cet algorithme de compression.