Compression de texte par la méthode du dictionnaire

Comprimer les données augmente la mémoire des ordinateurs et réduits le coût des communications.

Les logiciels de compression, rapides et faciles d'utilisation, permettent de diminuer de 50 pour cent ou plus la taille de quasiment tous les fichiers d'un ordinateur. Par exemple, pour quelques dizaines d'euros, des logiciels - dits doubleur de disque durs - transforment un disque dur en un disque de capacité double. Ou encore, le codage MP3 des données musicales, le codage JPEG des images (photos) et le codage MPEG des vidéos ont rendu possible le stockage et l'échange des fichiers par Internet.
Les techniques ou algorithmes de compression exploitent les régularités des données informatiques pour en diminuer la taille. Nous allons voir plus précisément sur deux exemples comment élucider ce miracle.

1 Deux exemples de techniques de compression

Expliquons comment, moyennant quelques conventions de codages bien choisies, on exploite les régularités des données informatiques pour en diminuer la taille.

1.1 Technique par répétitions locales

Imaginons que nous devons conserver un texte comportant des répétitions et ne contenant pas les symboles '[' et ']'. En convenant de mettre entre crochets les morceaux à répéter suivis du nombre de répétitions, on transformera le texte :

ainsi font font font les petites marionnettes ;
ainsi font font font les petites marionnettes ;

de 94 caractères (on compte les blancs), en un nouveau texte :

[ainsi [font]3 les petites marionnettes,]2 ;

qui ne comporte que 43 caractères.

Pour utiliser une telle méthode de codage on écrit deux algorithmes, l'un réalise la compression, c'est à dire fait passer du texte en clair au texte réduit, l'autre réalise la décompression, c'est à dire reconstitue le texte clair à partir du texte réduit.

1.2 Technique par référence ou adressage

Examinons un autre codage compressif. Dans une première partie du texte comprimé on enlève les séquences répétées et on écrit le texte restant en séparant chaque séquence par le symbole # (dont on suppose qu'il n'est pas utilisé dans le message à compresser). Puis dans une seconde partie, qui commence quand le symbole # est utilisé deux fois de suite, on indique quelles séquences doivent être utilisées et dans quel ordre. Notre message devient :

#ainsi #font #les petites marionnettes ;
##1#2#2#2#3#1#2#2#2#3 ;

Le message compressé contient 62 caractères. Le résultat est moins bon sur cet exemple mais la méthode s'applique dans un plus grand nombre de cas, car elle code les répétitions mêmes lorsqu'elles ne sont pas consécutives.

Cette seconde méthode ressemble d'ailleurs aux méthodes par dictionnaire qui font l'objet de cette séance d'informatique sans ordinateur.

Avant de passer aux exercices à destination des élèves, le lecteur notera que le contenu de cette page est extrait de 'L'intelligence et le calcul' de Jean-Paul Delahaye chez Belin|Pour la science. Nous conseillons fortement les ouvrages de cet auteur dans cette collection, dans lesquels le lecteur trouvera une description des grands principes de l'informatique théorique sous la forme d'une vulgarisation de très bon niveau.

2 Mise en pratique par les élèves

Dans le cadre de cette séance, nous proposons 5 fiches d'exercices.

2.1 Fiche 1 : 'un petit chat gris'

L'objectif de cet exercice est de montrer comment utiliser 'Le' dictionnaire pour compresser une comptine par la méthode de l'adressage. On utilise souvent une comptine comme support car ce type de chant comporte normalement un grand nombre de répétitions, même sur un texte très cours. La comptine retenue est :

Un petit chat gris.....19 car.
Un petit chat gris.....19 car.
Qui mangeait du riz.....20 car.
sur un tapis gris.....18 car.
Mais, ce n'est pas poli.....24 car.
De manger du riz.....17 car.
sur un tapis gris.....18 car.
Non, ce n'est pas poli.....23 car.
De manger du riz.....17 car.
sur un tapis gris.....18 car.

Cette comptine compte 193 caractères parmi lesquels les indications de passage à la ligne (parfois appelé 'retour chariot').

La technique de compression par adressage consiste à remplacer chaque mot par son 'adresse'. Il faut donc définir la notion d'adresse d'un mot. D'une façon simple, l'adresse d'un mot doit être une information qui permet de retrouver le mot, en évitant les ambiguïtés. La position d'un mot dans un dictionnaire, qui peut se définir comme un couple composé d'un numéro de page et d'une position dans cet page, nous semble bien adaptée pour définir une adresse. En supposant que le dictionnaire contient entre 100 et 999 pages et que chaque page contient moins de 100 mots, on peut faire l'hypothèse que l'adresse d'un mot prendra la forme (xyz/ab). Donc une adresse est composée de 8 caractères.

2.2 Fiche 2 : 'un petit chat gris' (la suite)

Cette solution pose néanmoins quelques problèmes puisque beaucoup de mots de la langue française n'apparaissent pas tels quels dans le dictionnaire. On peut citer le cas des conjugaisons possibles des verbes, des noms propres etc... (Donner d'autres exemples). Mais l'on peut imaginer que les algorithmes de compression et de décompression s'adaptent en ne retenant que les formes 'infinitifs' par exemple. Les caractères de ponctuations peuvent aussi rester sous une forme non compresser.

Sous une forme compressée la comptine ressemblerait à ça :

$\mathbf{adr_1 adr_2 adr_3 adr_4}$.....33 car.
$adr_1\, adr_2\, adr_3\, adr_4$.....33 car.
$adr_5\, adr_6\, adr_7\, adr_8$.....33 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$.....33 car.
$adr_{11}\,,\, adr_{12}\, adr_{13}\, adr_{14}\, adr_{15}\, adr_{16}$.....50 car.
$adr_{17}\, adr_6\, adr_7\, adr_8$.....33 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$.....33 car.
$adr_{18}\,,\, adr_{12}\, adr_{13}\, adr_{14}\, adr_{15}\, adr_{16}$.....50 car.
$adr_{17}\, adr_6\, adr_7\, adr_8$.....33 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$.....33 car.

La taille du texte compressé est de 364 caractères. Nous conviendrons donc que le gain est négatif. La raison principale de ce gain négatif est que nous compressons chaque mot par un couple dont la longueur est supérieure à la taille du mot lui même. Nous ne pouvons pas réduire la taille des mots à compresser. Par contre il est possible de réduire la taille d'une adresse. En effet :

- Le séparateur / à l'intérieur du couple n'est pas indispensable. Expliquer pourquoi.
- Les parenthèses ne sont pas nécessaires non plus. Expliquer pourquoi.

En appliquant ces deux remarques, la taille d'une adresse passe de 8 caractères à 5 caractères. Les tailles de chaque ligne de la comptine seront les suivantes :

$\mathbf{adr_1 adr_2 adr_3 adr_4}$ ..... 21 car.
$adr_1\, adr_2\, adr_3\, adr_4$ ..... 21 car.
$adr_5\, adr_6\, adr_7\, adr_8$ ..... 21 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$ ..... 21 car.
adr_{11}, $adr_{12}\, adr_{13}\, adr_{14}\, adr_{15}\, adr_{16}$ ..... 32 car.
$adr_{17}\, adr_6\, adr_7\, adr_8$ ..... 21 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$ ..... 21 car.
adr_{18}, $adr_{12}\, adr_{13}\, adr_{14}\, adr_{15}\, adr_{16}$ ..... 32 car.
$adr_{17}\, adr_6\, adr_7\, adr_8$ ..... 21 car.
$adr_9\, adr_1\, adr_{10}\, adr_4$ ..... 21 car.

La taille du texte compressé est de 232 caractères. Nous conviendrons encore que le gain reste négatif, même s'il se rapproche de 0. Expliquer quel est le problème fondamental auquel nous sommes confrontés. En d'autres mots, pourquoi la taille d'une adresse reste trop grande ? Comment la réduire de façon sensible ?

2.3 Fiche 3 : 'un petit chat gris' (suite et fin)

L'exercice précédent a montré que remplacer un mot par son adresse dans Le dictionnaire ne permet pas toujours de compresser un texte, surtout si les mots du texte sont courts. Le problème sous-jacent est que Le dictionnaire choisi contient tellement de mots (plusieurs dizaines de milliers) que l'adresse d'un mot devient longue. Il faut au moins trois caractères pour identifier une page parmi 1000, et deux autres caractères pour identifier un mot parmi 100.

La solution consiste à construire son propre dictionnaire qui contiendra seulement les mots du texte.

Adresses	*Mots*	Taille
01	*un*	5
02	*petit*	8
03	*chat*	7
04	*gris*	7
05	*qui*	6
06	*mangeait*	11
07	*du*	5
08	*riz*	6
09	*sur*	6
10	*tapis*	8
11	*mais*	7
12	*ce*	5
13	*n'est*	8
14	*pas*	6
15	*poli*	7
16	*de*	5
17	*manger*	9
18	*non*	6

TABLE 1 – Dictionnaire ad hoc pour la comptine du 'petit chat gris'. La dernière colonne donne la taille de chaque ligne du dictionnaire en tenant compte du passage à la ligne. Par exemple, la première ligne contient 2 caractères pour l'adresse, 2 pour le mot 'un', plus 1 pour le passage à la ligne. Nous obtenons un total de 122 caractères dans le dictionnaire.

Le texte de la comptine compressée :

01020304.....9 car.
01020304.....9 car.
05060708.....9 car.
09011004.....9 car.
11, *12131415*.....12 car.
16170708.....9 car.
09011004.....9 car.
18, *12131415*.....12 car.
16170708.....9 car.
09011004.....9 car.

La taille du texte compressé est de 96 caractères. Mais l'utilisation d'un dictionnaire ad hoc pose un problème, la personne, ou l'algorithme qui doit décompresser le texte doit disposer du dictionnaire. En effet, sans ce dictionnaire le texte n'est pas décodable. Il faut donc aussi transmettre le dictionnaire qui a une taille de 122 caractères, soit au total 218 caractères. Autrement dit nous avons choisi un texte pour lequel la méthode de compression n'est pas efficace. Deux raisons majeures à cela :

- Le texte est très court, le nombre de mots répétés n'est pas très important et ainsi la taille du dictionnaire lui même se rapproche de la taille du texte initial ;
- La taille moyenne des mots du texte est proche de 3 ou 4, donc il y a peu de différence entre la taille de l'adresse d'un mot dans le dictionnaire (2) et le mot lui même.

Nous proposons une dernière astuce, nous avons choisi de définir comme adresse un numéro dans une liste, comme nous avons plus de 10 mots dans le dictionnaire, et moins de 100, nous avons donc besoin de 2 caractères pour écrire chaque adresse. Mais puisque nous avons moins de 26 mots, nous pourrions adresser les mots du dictionnaire avec une simple lettre de l'alphabet.

On obtient le nouveau dictionnaire :

Adresses	*Mots*	Taille
a	*un*	4
b	*petit*	7
c	*chat*	6
d	*gris*	6
e	*qui*	5
f	*mangeait*	10
g	*du*	4
h	*riz*	5
i	*sur*	5
j	*tapis*	7
k	*mais*	6
l	*ce*	4
m	*n'est*	7
n	*pas*	5
o	*poli*	6
p	*de*	4
q	*manger*	8
r	*non*	5

TABLE 2 – Dictionnaire ad hoc pour la comptine du 'petit chat gris'. Chaque mot est indexé par une lettre. La taille du dictionnaire est 104.

La comptine compressée prend alors la forme suivante :

abcd.....5 car.
abcd.....5 car.
efgh.....5 car.
iajd.....5 car.
k, *lmno*.....7 car.
pqgh.....5 car.
iajd.....5 car.
r, *lmno*.....7 car.
pqgh.....5 car.
iajd.....5 car.

La taille du texte passe à 54 caractères au lieu des 96 caractères initialement, mais nous devons considérer aussi le dictionnaire dont la taille passe à 104 caractères. Ce qui donne un total de 158. Même s'il n'est pas très important, le gain est enfin positif.

2.4 Fiche 4 et Fiche 5

Ces fiches doivent être réalisées en petit groupe par les élèves (2*2 élèves par groupe). Chaque binôme doit mettre en application les principes étudiés sur un nouveau texte et vérifier si la méthode leur permet d'avoir un gain positif. Pour cela ils doivent construire un dictionnaire et compresser le texte. Ils échangent alors la fiche résultat avec le second binôme (qui n'aura pas eu le même texte) et doivent reconstruire le texte qui leur est transmis sous