

Compression des données

Exercices

**Titre****Objectif(s) :**

A la fin des exercices, l'élève doit être capable de :

- Comprendre ce qu'est la compression
 - Compression par dictionnaire
- Comprendre la compression RLE
- Comprendre le codage Huffman
- Comprendre la compression par dictionnaire LZ78
- Calculer le taux de compression
- Compression avec un outils comme 7Zip

Durée prévue : Selon UD

1 Compression par dictionnaire (exemple inventé)

Nous avons le texte ci-dessous à compresser qui à 59 caractères au total (1 caractères = 1 octet pour l'exemple) :

« un petit problème ou un tout petit problème est un problème »

Pour réaliser cet exemple de compression, nous allons créer un dictionnaire contenant chaque mot différent contenu dans cette phrase. Nous attribuerons un chiffre à chacun de ces mots, y compris l'espace. De cette manière nous arrivons à réduire le nombre d'octets nécessaire pour enregistrer le texte.

1) Établir un tableau :

- 1^{ère} colonne, le numéro de correspondance au mot
- 2^e colonne, chaque mot différent du texte
- 3^e colonne comptabiliser le nombre de symbole contenus dans les deux premières colonnes

Additionner la 3^{ème} colonne et mettre le total dans l'encadré

DICTIONNAIRE		Total caractères
Numéro	Mot unique	
1	un	3 (1un)
2	┐	2
	Nombre total de symboles	

2) Réécrire la phrase mais avec les numéros correspondants de la première colonne du tableau

La phrase devient donc (1 chiffre par case) :

[illegible]

Quel est le nombre de caractères de la phrase codée ?

- 3) Additionner le nombre de caractères figurant dans la phrase codée avec le total des caractères du dictionnaire :

Nb total de caractères du dictionnaire	
Nb caractères figurant dans la phrase codée	
Total	

- 4) Quel taux de compression obtenez-vous ?

2 Compression RLE (non destructive)

- 1) Décrire le principe RLE :

- 2) Sachant qu'une lettre est codée sur 1 octet, combien d'octets sont nécessaires pour coder sans compression WAAAAAOUUUUUUUUH ?

- 3) Coder avec cet algorithme la même chaîne de caractères (WAAAAAOUUUUUUUUH)

- 4) Sachant que, dans le code compressé, une lettre ou un chiffre seront aussi codés avec 1 octet, combien d'octets faut-il ?

- 5) Quel est le taux de compression sur cet exemple ?

- 6) Recommencer avec la chaîne de caractères suivante : SUPER

- 7) Taux de compression :

- 8) Conclusion :
Dans quel cas la compression par RLE est-elle intéressante ?

3 Codage Huffman (non-destructive)

L'algorithme de compression de Huffman est un système de codage statistique. Pour Huffman, il faut compter la fréquence de chaque caractère dans un texte ou une chaîne.

Voici le message à compresser "ta tata et ton tonton".

1) Pourquoi est-il faux de dire que ce message contient 17 caractères ?

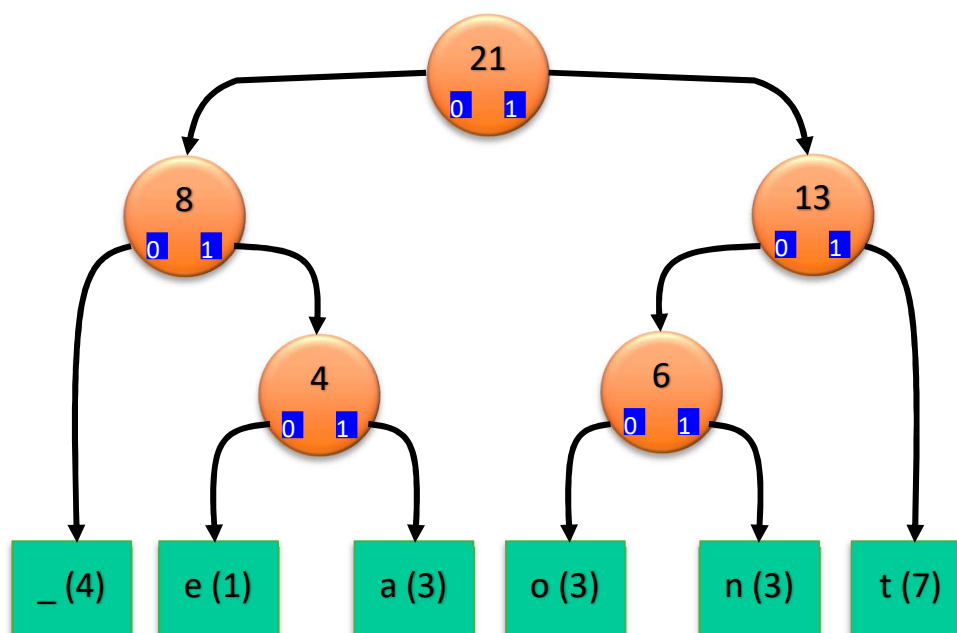
2) Combien en compte-t-il réellement ?

3) La méthode de Huffman se décompose en 3 étapes :

1ère étape : Compter les occurrences des différents caractères (_ pour espace) :

Lettres	t	_	a	o	n	e
Occurrences	7					

2^e étape : Construire l'arbre de codage.



On cherche à avoir des branches équilibrées.

Le code des caractères devient :

Lettres	t	-	a	o	n	e
Codes	11					

3^e étape :

Le message « **ta tata et ton tonton** » devient avec le code précédent (espacer les codes binaires) :

4) Calcul du taux de compression pour cet exemple :

Nombre de bits du message initial (8 bits par caractères) :

Nombre de bits du message final :

Taux de compression :

4 Compression par dictionnaire LZ78

Décoder le message suivant :

(0, v) (0, e) (0, r) (0, i) (0, d) (4, q) (0, u) (2, _) (7, n) (0, _) (0, p) (6, u) (8, n) (12, e)

Reconstruire le dictionnaire au fur et à mesure :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Message décodé :

Le dico contient éléments mais on compte jusqu'àmax. pour indiquer la position dans le dictionnaire, ce nombre se code sur bits.

La phrase initiale faisait caractères soit bits.

La phrase codée fait bits.

Soit un taux de compression de

5 Compression des fichiers archives

5.1 Le format .docx

Avec Word, créez un nouveau document et le nommer « **document.docx** ». Ecrire « **Hello** », puis le fermer. Changer l'extension du document en « **zip** », et le dézipper avec 7-zip. Rechercher dans les fichiers le texte « **Hello** » puis le renommer avec le texte « **Salut** ». Récrire l'archive avec 7-zip, et changer l'extension en « **docx** ». L'ouvrir avec Word.

Que constatez-vous ?

5.2 Les dossiers compressés

Récupérez le fichier « sauvegarde.exe » et exécutez-le afin d'extraire dans un dossier de travail les fichiers nécessaires. Le dossier « sauvegarde » sera créé automatiquement.

Créer une archive « **zip** » depuis l'explorateur de fichier (il gère les zip) du dossier « **sauvegarde** ». Si vous ne savez pas comment faire, un clic droit sur le dossier devrait vous aider.

Pourquoi est-ce important de faire cela si l'on veut pérenniser les données de ce dossier ?

5.3 7-Zip

Récupérez « paysage.exe » procédez à son auto extraction dans votre dossier de travail.

Cette fois, **avec l'application 7-Zip** faites les manipulations ci-dessous.

Faites une archive de chaque fichier séparément (bmp.7z et png.7z) selon leur extension avec les paramètres suivants :

- Format de l'archive : 7z
- Niveau de compression : 9 - Ultra

Pourquoi « png.7z » a quasi le même poids que « paysage.png » à quelques octets près ?

Que s'est-il passé pour le fichier « paysage.bmp » ?

Suite de l'exercice sur la page suivante...

⚠ Supprimez les deux archives 7z avant de procéder à la suite !

Maintenant faite une archive du dossier « paysage » avec les 2 images uniquement en utilisant les paramètres suivants :

- Format de l'archive : 7z
- Niveau de compression : 5 - Normale
- Chiffrement, mot de passe : toto123
- Diviser en volumes, octets : 1M

Il y a maintenant 4 fichiers qui composent cette archive, pourquoi et à quoi cela peut-il servir ?

Pourquoi le mot de passe ?

6 Calcul du poids d'un audio

Démontrez par calcul pourquoi le fichier « musique.wav » fait 3.36 Mo :

- 16 bits
- 44 100 Hz
- Stéréo (2 voies)
- Durée 20"