L'esprit en éveil : 25 ans de culture scientifique

PRINTEMPS DES SCIENCES 23-29 MARS 2026

Compresse-moi si tu peux !

ULB UNIVERSITÉ LIBRE DE BRUXELLES

UNIVERSITÉ LIBRE DE BRUXELLES – FACULTÉ DES SCIENCES

DÉPARTEMENT D'INFORMATIQUE

Thamid Rahman, Alae Ben Messaoud, Min-Tchun Li, Casper Suteniec et Meryeme Lachhab

UNIVERSITÉ LIBRE DE BRUXELLES – FACULTÉ DES SCIENCES

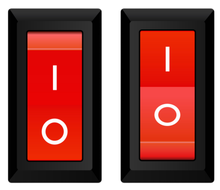
Comment mettre tous ces t-shirts dans cette armoire ?

Comment mettre tous ces vêtements dans l'armoire ?

Comment stocker les données dans un espace limité ?

- **Binary Digit (ou Bit)**: l'ordinateur ne comprend que

2 états : Éteint (0) ou Allumé (1)



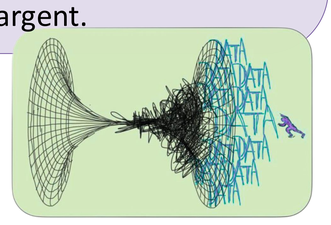
- **Système binaire** : langage formé de la combinaison de plusieurs bits
- **encodage** : suite de 8 bits **(un Octet)** = une lettre

Par exemple : 01000001 = lettre A .

Pourquoi ne pas juste tout balancer dans l'armoire ?

vêtements en vrac : armoire bloquée ⟷ données brutes : serveurs saturés

«plier » les données : économie d'espace et d'argent.



- **entropie de Shannon (H)** : limite de la compression, mesure du désordre.
- **entropie maximale** : si les données sont totalement aléatoires, l'entropie est au maximum : compression impossible sans pertes.

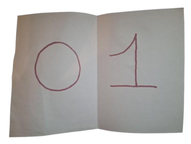
2 types de compression :

SANS PERTE	AVEC PERTE
indispensable pour textes et logiciels (perte de bit corrompt les données)	utilisé pour le multimédia
respecte limite de Shannon	suppression des détails invisibles/inaudibles
Exemple : .PNG, .ZIP	Exemple : .JPEG, .MP3

Hello World !
H???o W??ld

Compresser des données, c'est comme (regrouper?) plier des t-shirts

- **But de la compression** : Recherche + élimination de la redondance



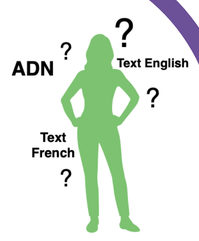
- **taux de compression** : formalise l'efficacité de compression par

$$R = \frac{\text{Taille non-compressé}}{\text{Taille compressé}}$$

- **Exemple** :
  - Au lieu de dire à l'ordinateur : "Pixel 1 : Bleu" "Pixel 2 : Bleu" "Pixel 3 : Bleu..."
  - L'algorithme de compression va noter : "Les 500 prochains pixels sont Bleus."

Aucune information supprimée !

Notre expérience : Comment « plier » des données textuelles ?

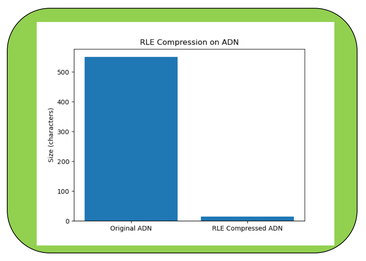


- **Approche expérimentale** :
  1. comparaison de plusieurs méthodes de compression
  2. différents types de données : texte, ADN, chiffres.
- **Analyse des résultats**

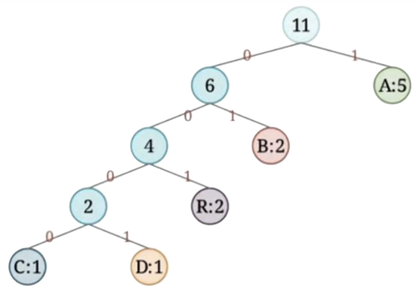
1. RLE
- Efficace sur données très redondantes (ex: ADN)
- code les répétitions consécutives

AAAAA → A5

ACGTACGT → 1A1C1G1T1A1C1G1T



2. Huffman
- utilisation d'un arbre binaire
- code + court attribués aux caractères + fréquents



- ABRACADABRA :
- A -> 1 R -> 01 B -> 001 C -> 0000 D -> 0001

3. Compromis : - En théorie RLE + rapide, mais efficacité limitée !

- Huffman plus lent, mais portée de données plus grande

Si aucune lettre ne se suit (ex: ACGT), il peut doubler le temps de compression au lieu de le réduire

CONCLUSION : des jeans ne se plient pas de la même manière qu'un t-shirt

pliage optimal dépend de la **nature du vêtement** ⟷ compression optimale dépend de la **nature des données**

Ce qu'il faut retenir :

- **Choisir la bonne méthode = économiser de l'espace !**
- **Compromis fondamental** :

Pas de solution miracle au problème : il faut trouver un compromis

Temps de calcul ⟷ Gain d'espace (taux de compression)

- **Exemple** : algorithme de Huffman performant sur du texte MAIS inefficace sur des données aléatoire (car redondance y est minimale)