



© César, Musée d'Art Contemporain, Marseille
Photo: markovdz sur flickr

Information, Calcul et Communication

Compression de données : Introduction

Olivier Lévêque

EPFL Pourquoi donc vouloir compresser des données ?

Deux raisons principales :

- Pour réduire l'espace utilisé lors du **stockage** de ces données
- Pour réduire le temps de **transmission** de ces données

Quels types de données peuvent être compressées ?

- Les textes
- Les sons
- Les images
- Les vidéos
- En général, tout type de données numériques !

Deux types de compression

- **Compression sans pertes** : lorsqu'on désire retrouver l'intégralité des données stockées sous forme compressée
- **Exemples** : billets pour un concert, bulletins de vote, articles scientifiques
- **Compression avec pertes** : lorsqu'on n'est pas tant à cheval que ça sur les détails et qu'on s'autorise un peu de **distorsion**
- **Exemples** : morceaux de musique au format mp3, partage de photos sur le web, vidéos YouTube...

- Aussi étrange que cela puisse paraître, il est possible de réduire la taille d'un fichier informatique...
- ...sans pour autant perdre la moindre information à propos du fichier !
- L'idée de base consiste à **supprimer/réduire la redondance** présente dans les données en **abrégéant les motifs qui reviennent souvent** dans celles-ci.

La langue française est pleine de redondance !

Pour preuve, voici deux phrases :

~~Cette première phrase dont les lettres ont été
à moitié effacées est encore parfaitement lisible.~~

Si on liasse la perimère et la drnièree ltretes à la
bnnoe pclae dnas cquahe mot, arols ctete sncedone
prshae est asusi pfatitaerment liiblse (ou psqeure!)

Pourquoi tant de redondance ?

- pour mieux se comprendre, tout simplement !
- pour être capable de lire un texte contenant des fotes d'ortografe.

Exemples de compression dans la vie de tous les jours

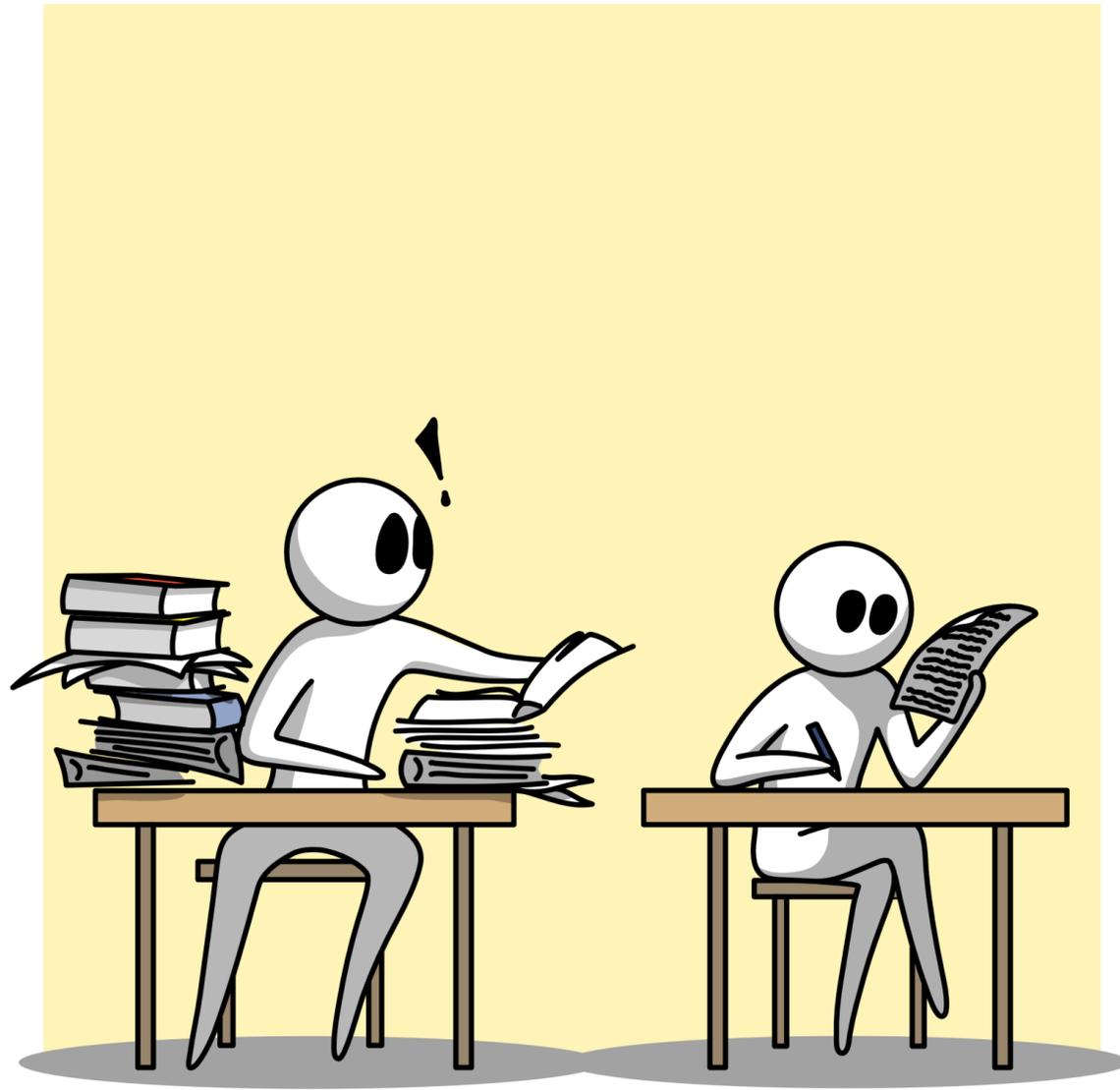
- **Langage SMS** : “slt”, “tqt”, “mdr”, ...
- **Code Morse** : A = “.-”, E = “.”, S = “...”, T = “-”,
tandis que X = “-..-”, Z = “--..”
- **Acronymes** : EPFL, UNIL, ...

Exemple de compression (sans pertes)

Essayons d'encoder les séquences de lettres **MONTREUX** et **LAUSANNE** sous la forme de séquences de 0 et de 1 :

- La séquence **MONTREUX** a huit lettres différentes : il n'y a donc pas de choix pour cette séquence : 3 bits par lettre sont nécessaires (par exemple : **M** ↔ 000, **O** ↔ 001, etc.) et donc $3 \times 8 = 24$ bits en tout.
- Dans la séquence **LAUSANNE** par contre, les lettres **A** et **N** se répètent chacune deux fois ; on peut donc abrégier la représentation de celles-ci en utilisant le code suivant (par exemple) : **A** ↔ 11, **N** ↔ 10, **L** ↔ 000, **S** ↔ 001, **U** ↔ 010, **E** ↔ 011.

Cette représentation n'utilise que $2 \times 4 + 3 \times 4 = 20$ bits en tout.



Information, Calcul et Communication

Entropie

Olivier Lévêque

Exemple 1

Voici une séquence de 16 lettres :

A B C D E F G H I J K L M N O P

Jeu n° 1 :

Deviner quelle lettre a été tirée au hasard en posant un nombre minimum de questions binaires, auxquelles on ne répond que par oui ou par non.

Q1: A-H ? oui ✓ 8 poss. à G, 8 poss. à D

Q2: C-G ? oui

Q2': A-D ? oui

Q3: E ? non

Q3': A-B ? non

Q4: D ? oui ✓

Q4': C ? non

Exemple 1

Voici une séquence de 16 lettres :

A B C D E F G H I J K L M N O P

Jeu n° 1 :

Deviner quelle lettre a été tirée au hasard en posant un nombre minimum de questions binaires, auxquelles on ne répond que par oui ou par non.

Solution :

- 4 questions sont nécessaires (algorithme de dichotomie).
- On dit que l'entropie de cette séquence est égale à 4.
- Remarquez que $16 = 2^4$, autrement dit : $4 = \log_2(16)$

EPFL Exemple 2

Voici une autre séquence de 16 lettres (sans compter les espaces):

IL FAIT BEAU A IBIZA

Jeu n° 2 : Le jeu est le même qu'avant !

Remarques:

- La position de la lettre est tirée au hasard, de manière uniforme.
- Il ne faut deviner que la lettre elle-même (I, L, F, A, ...), pas sa position.

Q1: I, B, A ? oui

Q1': A ou I ? oui

Q2: A ? oui

Q2': A ? oui

EPFL Exemple 2

Voici une autre séquence de 16 lettres (sans compter les espaces):

IL FAIT BEAU A IBIZA

Jeu n° 2 : Le jeu est le même qu'avant !

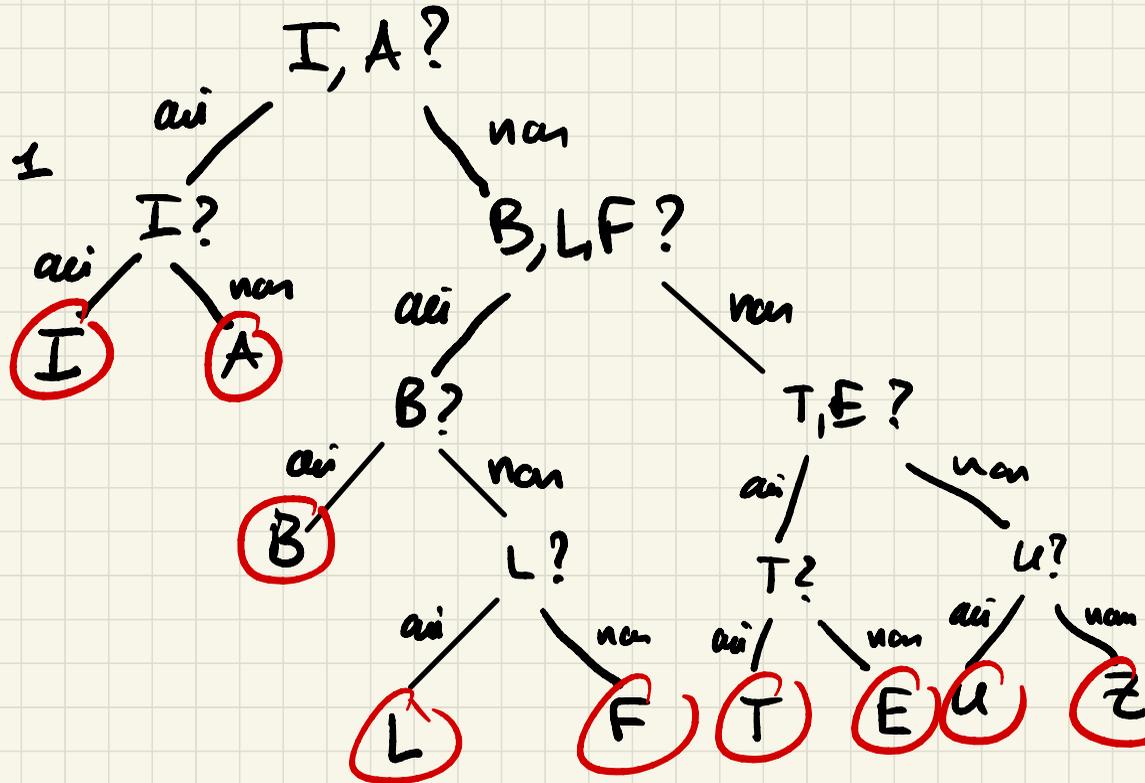
Remarques:

- La position de la lettre est tirée au hasard, de manière uniforme.
- Il ne faut deviner que la lettre elle-même (I, L, F, A, ...), pas sa position.

Combien de questions binaires *en moyenne* sont-elles nécessaires pour deviner la lettre ?

IL FAIT BEAU A IBIZA

I	4	2
A	4	
B	2	3
L	1	4
F	1	2
T	1	4
E	1	3
U	1	4
Z	1	4



EPFL Exemple 2

IL FAIT BEAU A IBIZA

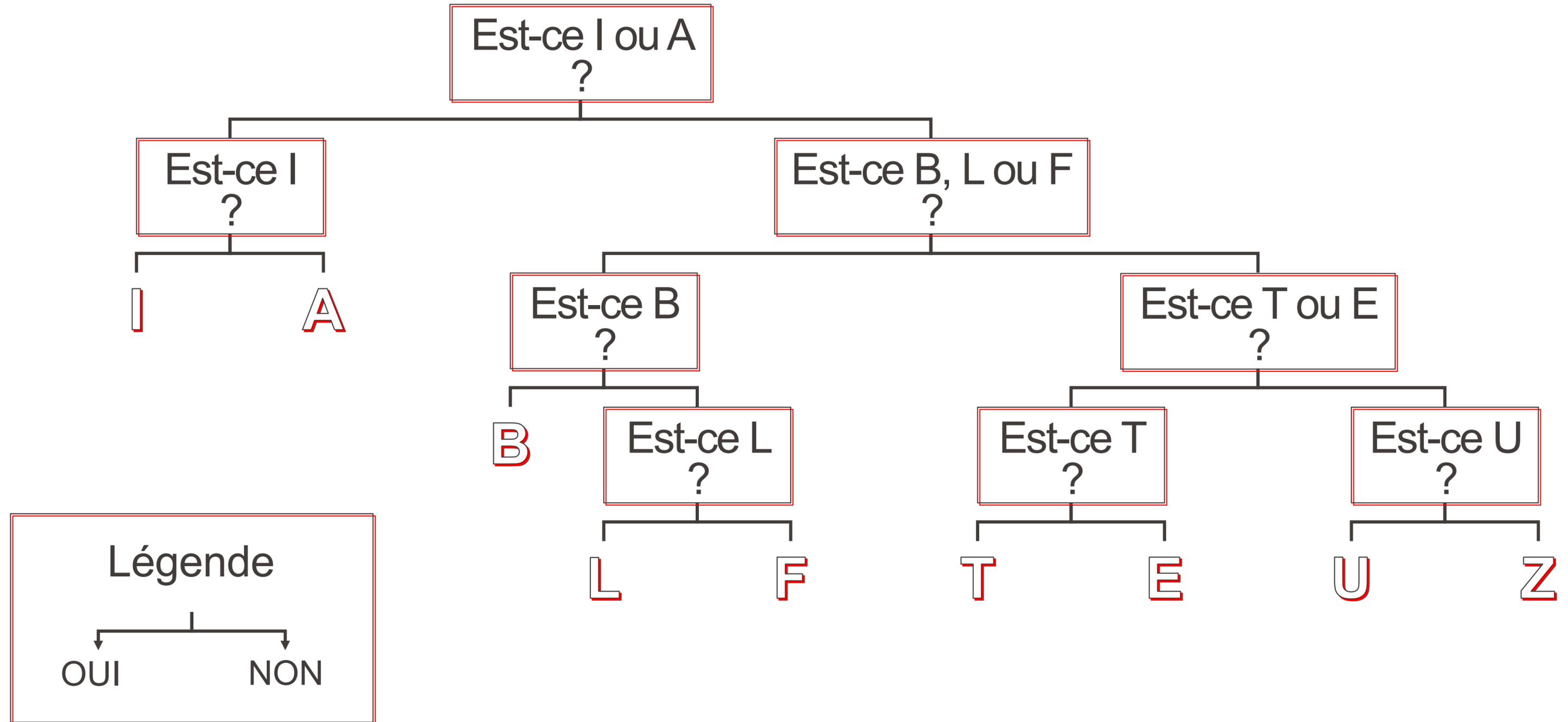
Solution: classer les lettres dans l'ordre **décroissant** du nombre d'apparitions dans la séquence :

Lettre	I	A	B	L	F	T	E	U	Z
Nombre d'apparitions	4	4	2	1	1	1	1	1	1

Idée (dichotomie plus générale) : séparer l'ensemble des lettres en deux parties égales en tenant compte de leur **nombre d'apparitions**, ce qui donne :

- **Question n° 1 :** Est-ce que la lettre est **un I ou un A** ?
 - Si la réponse est **oui** : **Question n° 2:** Est-ce que la lettre est **un I** ?
 - Si la réponse est **non** : **Question n° 2:** Est-ce que la lettre est **un B, L ou F** ?
- etc.

Exemple 2 : Arbre des questions



Exemple 2 : Solution

Lettre	I	A	B	L	F	T	E	U	Z
Nombre d'apparitions	4	4	2	1	1	1	1	1	1
Nombre de questions	2	2	3	4	4	4	4	4	4

Nombre de questions à poser en moyenne :

$$= 2 \cdot \frac{4}{16} \cdot 2 + 1 \cdot \frac{2}{16} \cdot 3 + 6 \cdot \frac{1}{16} \cdot 4 = \frac{16 + 6 + 24}{16} = \frac{46}{16} = 2.875$$

On dit que **l'entropie** de cette séquence est égale à **2.875**.

Exemple 3

Voici encore une autre séquence de 16 lettres :

A A A A A A A A A A A A A A A A

Jeu n° 3: Le jeu est encore le même qu'avant !

- Cette fois-ci, **aucune** question n'est nécessaire pour deviner la lettre choisie!
- On dit que **l'entropie** de cette séquence est égale à **0**.

Retour à l'exemple 2

Lettre	I	A	B	L	F	T	E	U	Z
Nombre d'apparitions	4	4	2	1	1	1	1	1	1
Nombre de questions	2	2	3	4	4	4	4	4	4

Remarques :

- Pour deviner une lettre qui apparaît 1 fois sur 16, on a besoin de 4 questions.
 $4 = \log_2(16)$
- Pour deviner une lettre qui apparaît 2 fois sur 16 (i.e. 1 fois sur 8), on a besoin de 3 questions. $3 = \log_2(8)$
- Pour deviner une lettre qui apparaît 4 fois sur 16 (i.e. 1 fois sur 4), on a besoin de 2 questions. $2 = \log_2(4)$

En résumé : pour deviner une lettre qui apparaît avec une probabilité p ,
on a besoin de $\log_2\left(\frac{1}{p}\right)$ questions.

Entropie : Définition générale

- Soit X une séquence de lettres provenant d'un alphabet $A = \{a_1, \dots, a_n\}$.
- Soit p_j la probabilité d'apparition de la lettre a_j dans la séquence X (remarquez que $0 \leq p_j \leq 1 \forall j$ et que $p_1 + \dots + p_n = 1$).

L'entropie de la séquence X est définie par :

$$H(X) = p_1 \cdot \log_2 \left(\frac{1}{p_1} \right) + \dots + p_n \cdot \log_2 \left(\frac{1}{p_n} \right)$$

Remarque : par convention, si $p_j = 0$, alors on pose $p_j \cdot \log_2 \left(\frac{1}{p_j} \right) = 0$.

Entropie : Exemple

Lettre	I	A	B	L	F	T	E	U	Z
Nombre d'apparitions	4	4	2	1	1	1	1	1	1
Nombre de questions	2	2	3	4	4	4	4	4	4

$$H(X) = 2 \cdot \frac{1}{4} \cdot \log_2(4) + 1 \cdot \frac{1}{8} \cdot \log_2(8) + 6 \cdot \frac{1}{16} \cdot \log_2(16) = 2.875$$

Notez qu'en général, l'entropie **ne coïncide pas** avec le nombre moyen de questions binaires à poser pour deviner une lettre !

Exemple : Trouver une lettre tirée au hasard dans la séquence **ABB** nécessite exactement 1 question binaire, mais

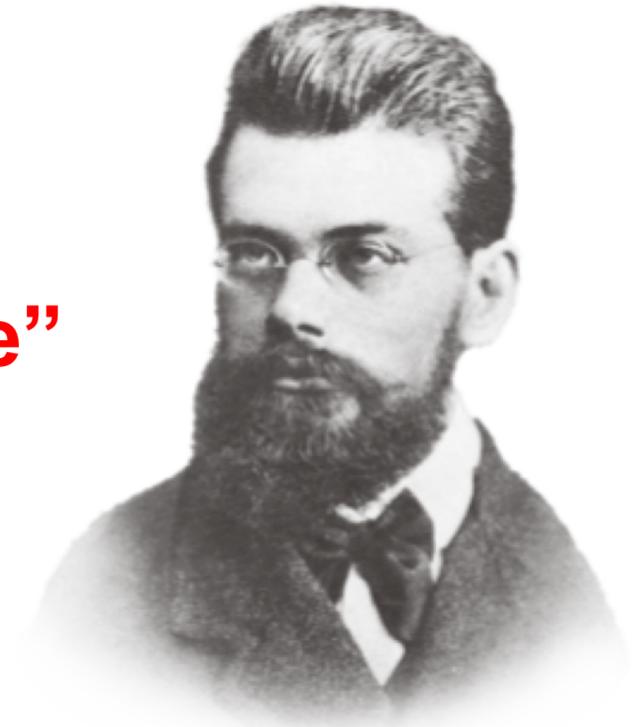
$$H(X) = \frac{1}{3} \cdot \log_2(3) + \frac{2}{3} \cdot \log_2\left(\frac{3}{2}\right) \simeq 0.92$$

Origines de la notion d'entropie

Origine en physique (Boltzmann, 1872) :

L'entropie mesure le **“désordre”**
dans un système physique.

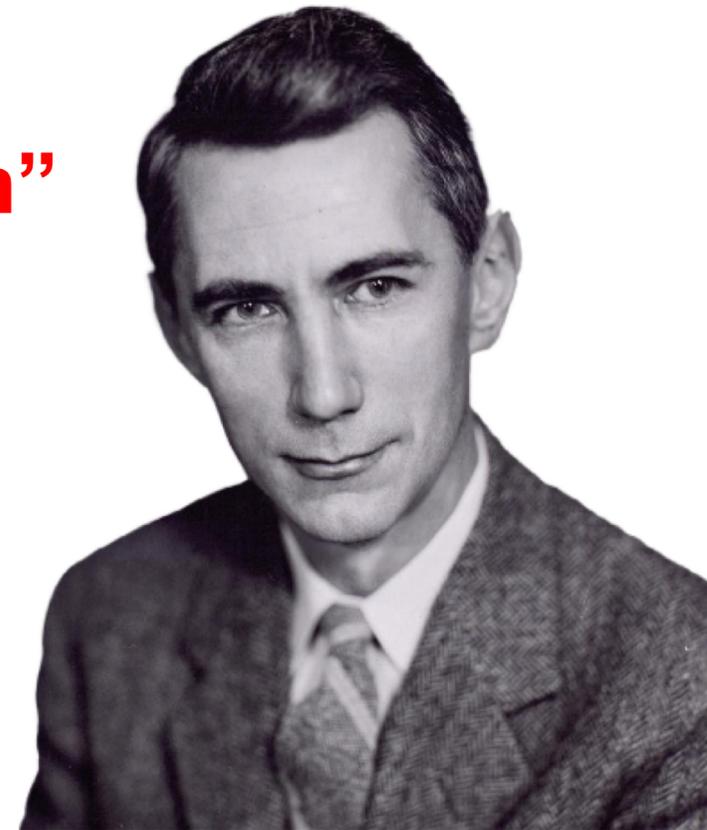
- Ludwig Boltzmann (1844-1905)
- ardent défenseur de l'existence des atomes
- père de la physique statistique



Théorie de l'information (Shannon, 1948) :

L'entropie mesure la **“quantité d'information”**
contenue dans un signal.

- Claude Shannon (1916-2001)
- mathématicien, ingénieur électricien, cryptologue,
- père de la théorie de l'information, jongleur...



Interprétation de la notion d'entropie

A B C D E F G H I J K L M N O P $H(X) = 4$

IL FAIT BEAU A IBIZA $H(X) = 2.875$

A A A A A A A A A A A A A A A $H(X) = 0$

- Plus il y a de lettres différentes, plus il y a de désordre, plus il y a de **nouveauté** et donc “**plus d’information**” dans le message.
- Plus il y a de lettres semblables, moins il y a de désordre, plus il y a de **redondance** et donc “**moins d’information**” dans le message.

Quelques propriétés de l'entropie

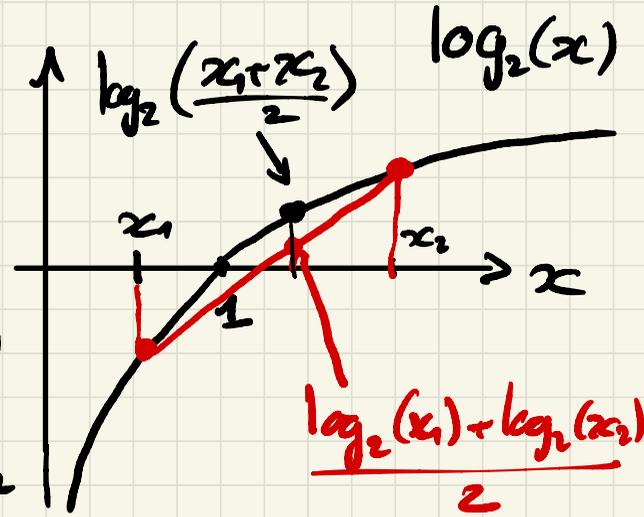
$$H(X) = p_1 \cdot \log_2 \left(\frac{1}{p_1} \right) + \cdots + p_n \cdot \log_2 \left(\frac{1}{p_n} \right)$$

- Pour une probabilité d'apparition $0 \leq p \leq 1$ donnée, $\log_2 \left(\frac{1}{p} \right) \geq 0$.
- $H(X) \geq 0$ en général, et $H(X) = 0$ si et seulement si l'ordre est total (c'est-à-dire si toutes les lettres sont les mêmes).
- Si n est la taille de l'alphabet utilisé, $H(X) \leq \log_2(n)$ en général et $H(X) = \log_2(n)$ si et seulement si le désordre est total (c'est-à-dire si toutes les lettres sont différentes).

$$\underline{H(x) \leq \log_2 n:}$$

$$\frac{1}{2}(\log_2(x_1) + \log_2(x_2)) \leq \log_2\left(\frac{x_1 + x_2}{2}\right)$$

Concavité de \log_2



$$p_1 \log_2(x_1) + p_2 \log_2(x_2) \leq \log_2(p_1 x_1 + p_2 x_2) \quad p_1 + p_2 = 1$$

$$p_1 \log_2(x_1) + \dots + p_n \log_2(x_n) \leq \log_2(p_1 x_1 + \dots + p_n x_n)$$

$$p_1, \dots, p_n \geq 0, \quad p_1 + \dots + p_n = 1$$

$$p_1 \log_2(x_1) + \dots + p_n \log_2(x_n) \leq \log_2(p_1 x_1 + \dots + p_n x_n)$$

$$p_1, \dots, p_n \geq 0, p_1 + \dots + p_n = 1$$

$$H(x) = p_1 \log_2\left(\frac{1}{p_1}\right) + \dots + p_n \log_2\left(\frac{1}{p_n}\right) \quad x_i = \frac{1}{p_i}$$

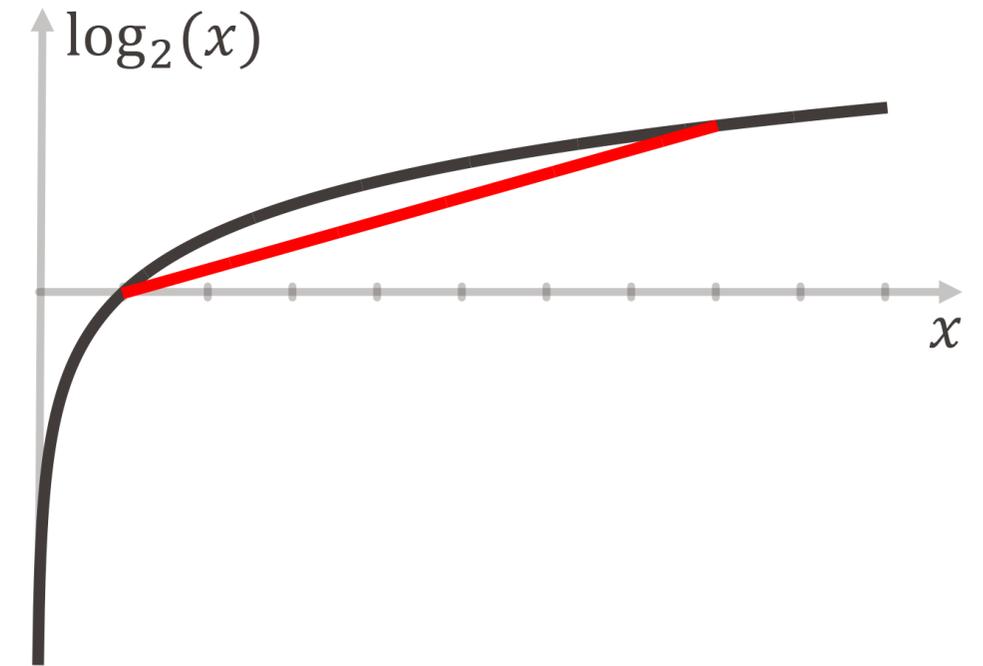
$$\leq \log_2\left(p_1 \cdot \frac{1}{p_1} + \dots + p_n \cdot \frac{1}{p_n}\right)$$

$$= \log_2(1 + \dots + 1) = \log_2(n) \quad \#$$

Quelques propriétés de l'entropie

Vérifions que $H(X) \leq \log_2(n)$ en général :

- Remarquez que la fonction $f(x) = \log_2(x)$ est concave pour $x \geq 0$



- En particulier, ceci veut dire que :

$$\frac{\log_2(x_1) + \log_2(x_2)}{2} \leq \log_2\left(\frac{x_1 + x_2}{2}\right) \quad \forall x_1, x_2 \geq 0$$

- Plus généralement, si $0 \leq p_1, p_2 \leq 1$ et $p_1 + p_2 = 1$, alors :

$$p_1 \cdot \log_2(x_1) + p_2 \cdot \log_2(x_2) \leq \log_2(p_1 x_1 + p_2 x_2) \quad \forall x_1, x_2 \geq 0$$

Quelques propriétés de l'entropie

Vérifions que $H(X) \leq \log_2(n)$ en général : (suite)

- Plus généralement encore, si $0 \leq p_j \leq 1$ et $p_1 + \dots + p_n = 1$, alors

$$p_1 \cdot \log_2(x_1) + \dots + p_n \cdot \log_2(x_n) \leq \log_2(p_1 x_1 + \dots + p_n x_n) \quad \forall x_1, \dots, x_n \geq 0$$

- En appliquant cette inégalité avec $x_j = \frac{1}{p_j}$, on obtient finalement :

$$H(X) = p_1 \cdot \log_2\left(\frac{1}{p_1}\right) + \dots + p_n \cdot \log_2\left(\frac{1}{p_n}\right) \leq \log_2\left(\frac{p_1}{p_1} + \dots + \frac{p_n}{p_n}\right) = \log_2(n)$$



IL FAIT BEAU A IBIZA

16 lettres
(~~espaces~~)

But: encoder cette séquence sans forme de 0 et de 1 en utilisant le moins de bits possible.

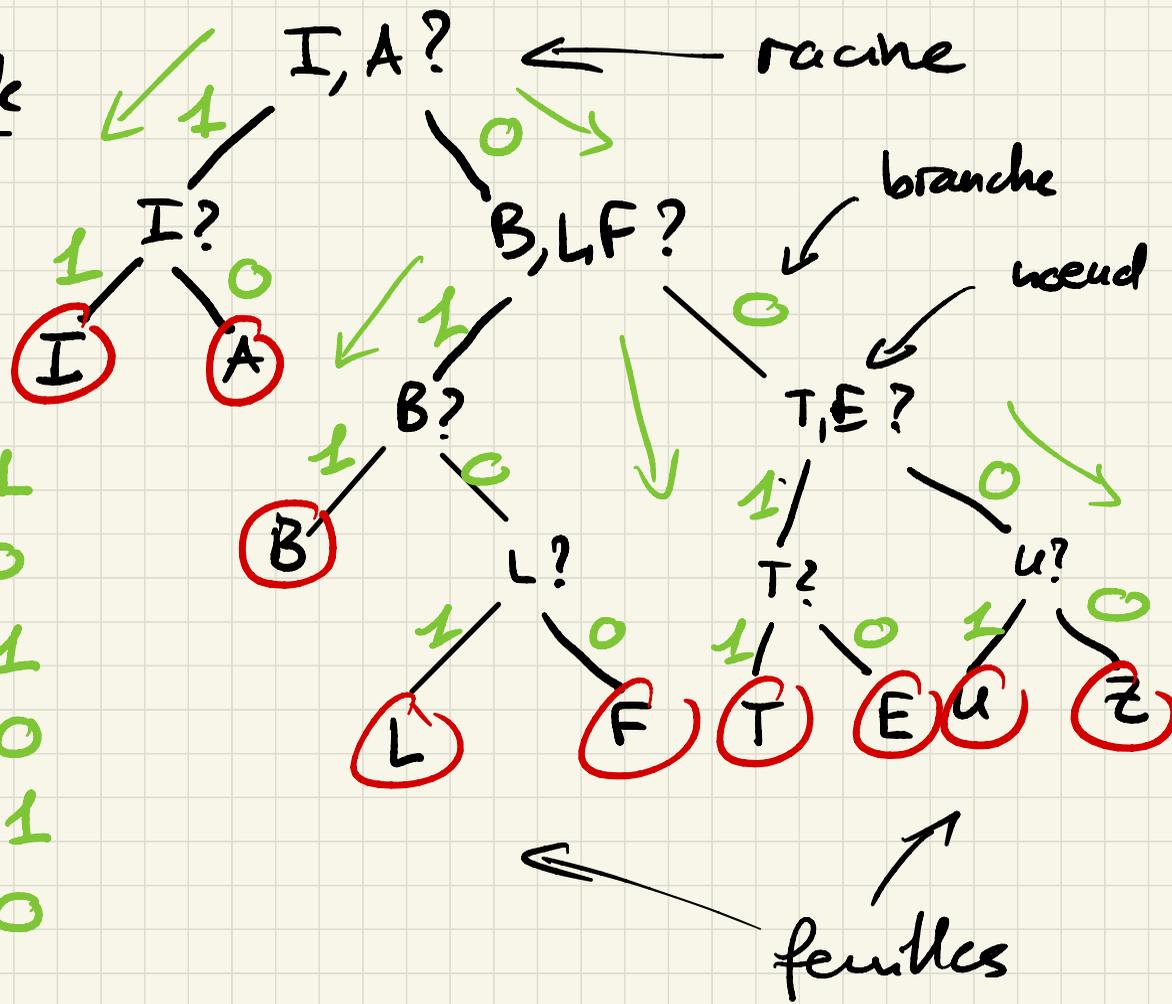
Idée 0: utiliser code ASCII; 8 bits / lettre

Idée 1: utiliser un nb fixe de bits / lettre: 4
(9 lettres différentes)

Idée : proposer un système d'encodage tel que :

- le nombre de bits associé à chaque lettre = nb de questions nécessaires pour deviner celle-ci
- réponse oui $\leftrightarrow 1$
réponse non $\leftrightarrow 0$

lettre	mot de code
I	11
A	10
B	011
L	0101
F	0100
T	0011
E	0010
U	0001
Z	0000



Nombre total de bits utilisés:

$$\begin{array}{ccccccc} 8 \cdot 2 & + & 2 \cdot 3 & + & 6 \cdot 4 & = & 16 + 6 + 24 = 46 \\ \text{I, A} & & \text{B} & & \text{L F T E U Z} & & \text{bits} \end{array}$$

16 lettres au total dans le message

→ on a besoin de $\frac{46}{16} = 2,875$ bits / lettre

(en général, \geq) = entropie!

IL FAIT BEAU ...

encodage: $\frac{11}{I} | \frac{0101}{L} | \frac{0100}{F} | 10011 \dots$

décodage: il suffit de lire de gauche à droite!

A	00
B	01
C	100
D	10

ACDC

00 | 100 10 100

1°) A D B B A ?

2°) A C D C

Pb = D est le préfixe de C

Question

Si on a un fichier de données ?

0110110111000110001...
└──────────┬──────────┘
1^{er} octet 2^e octet