Algorithmique des structures arborescentes

L2 Info et Math-info, 2017–18

Marc Zeitoun

28 mars 2018

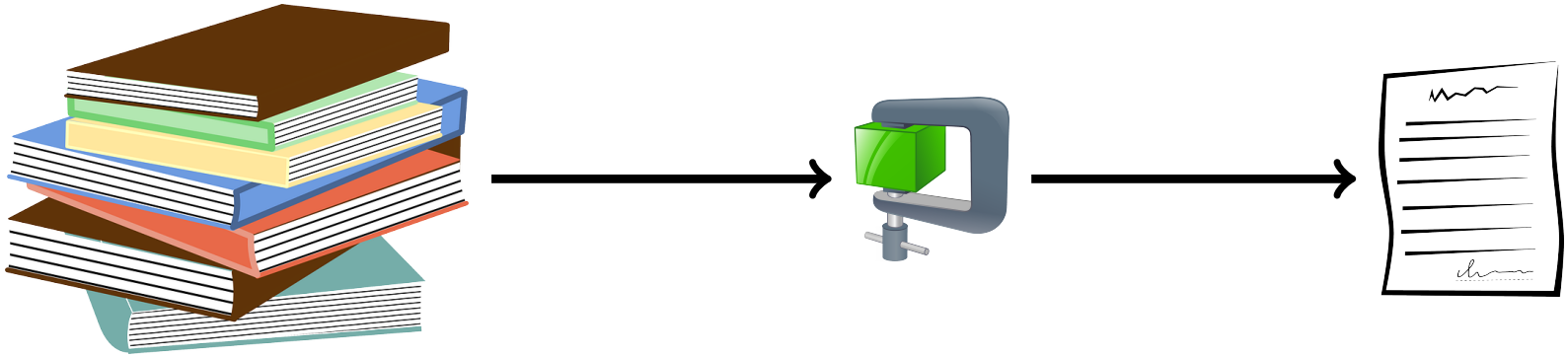
Organisation

- Aujourd'hui = dernier cours en amphi.
- DS demain 8h-9h15.

Il est conseillé d'arriver à 7h45.

Plan

Compression sans perte

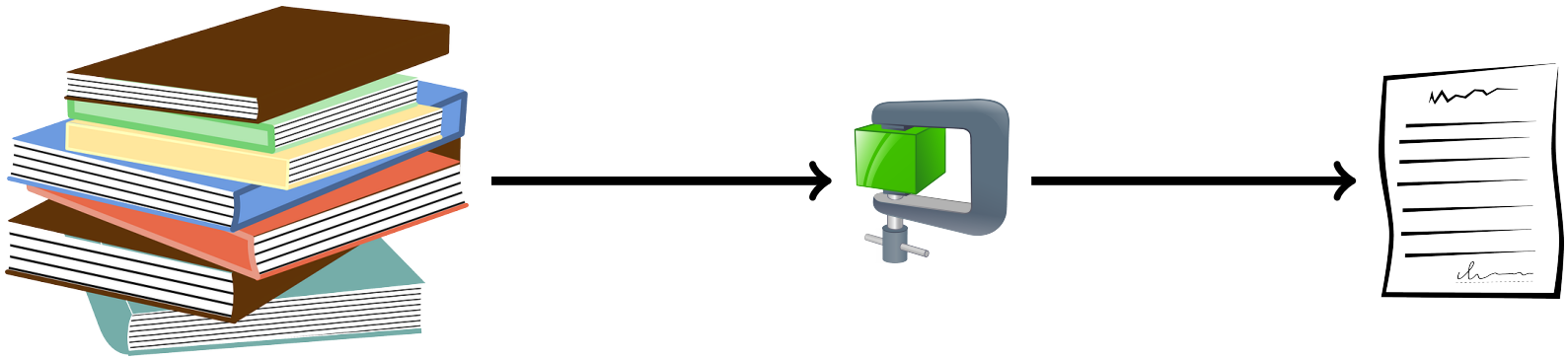


1Tb **25Kb**

Objectifs :

- Réduire la taille des fichiers.

Compression sans perte

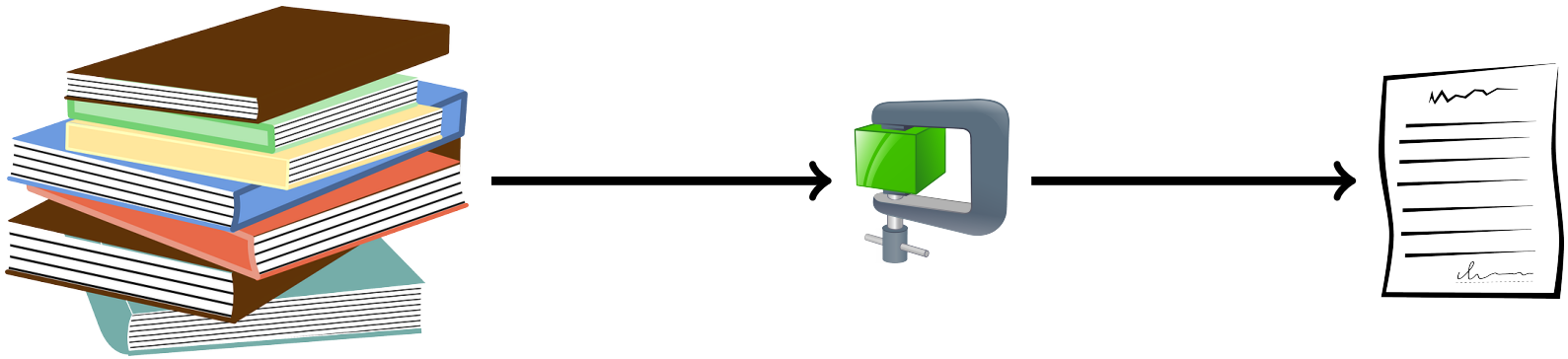


1Tb **25Kb**

Objectifs :

- Réduire la taille des fichiers.
- **Sans perte** : on veut pouvoir reconstruire le fichier **original**.

Compression sans perte



1Tb **25Kb**

Objectifs :

- Réduire la taille des fichiers.
- **Sans perte** : on veut pouvoir reconstruire le fichier **original**.
- Aujourd'hui : 2 algorithmes : LZ78 et Huffman.
  Implémentation utilisant des arbres binaires.

Codages de caractères

- Un fichier sur disque est une suite de caractères.

Codages de caractères

- Un fichier sur disque est une suite de caractères.
- Un codage associe à chaque caractère une **représentation** par un ou plusieurs **octets**.

Codages de caractères

- Un fichier sur disque est une suite de caractères.
- Un codage associe à chaque caractère une **représentation** par un ou plusieurs **octets**.
- Il y a plusieurs codages existants. Par exemple,
  - ASCII : représente 128 caractères chacun **sur un octet**.

Codages de caractères

- Un fichier sur disque est une suite de caractères.
- Un codage associe à chaque caractère une **représentation** par un ou plusieurs **octets**.
- Il y a plusieurs codages existants. Par exemple,
  - ASCII : représente 128 caractères chacun **sur un octet**.
  - ISO 8859-1 (latin1) l'étend à 191 caractères **sur un octet**.

Codages de caractères

- Un fichier sur disque est une suite de caractères.
- Un codage associe à chaque caractère une **représentation** par un ou plusieurs **octets**.
- Il y a plusieurs codages existants. Par exemple,
  - ASCII : représente 128 caractères chacun **sur un octet**.
  - ISO 8859-1 (latin1) l'étend à 191 caractères **sur un octet**.
  - UTF-8 pour les caractères du standard Unicode : 1 à 4 octets.
    $\Longrightarrow$ Plus difficile de décoder un fichier UTF-8.

Plan

Compression de Huffman : principe

- Idée : coder les caractères sur un **nombre variable de bits**, éventuellement moins que 8.
- Caractères **fréquents** représentés par des codes **courts**.
- Caractères **peu fréquents** représentés par des codes **longs**.
- **Difficulté** trouver un codage permettant de décoder.
- Par exemple, si on code le caractère **a** par **0** et **b** par **00**, on ne pourra pas décoder la suite 000 : elle peut représenter ab ou ba.

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.
2. Calcul du codage de chaque caractère, avec 2 contraintes :

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.
2. Calcul du codage de chaque caractère, avec 2 contraintes :
   - caractères les plus fréquents représentés par des codes **courts**,

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.
2. Calcul du codage de chaque caractère, avec 2 contraintes :
   - caractères les plus fréquents représentés par des codes **courts**,
   - possibilité de décoder.

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.
2. Calcul du codage de chaque caractère, avec 2 contraintes :
   - caractères les plus fréquents représentés par des codes **courts**,
   - possibilité de décoder.
3. Dans le fichier compressé, écrire le codage utilisé.

Compression de Huffman : étapes

1. Calcul des fréquences de caractères dans le fichier à compresser.
2. Calcul du codage de chaque caractère, avec 2 contraintes :
   - caractères les plus fréquents représentés par des codes **courts**,
   - possibilité de décoder.
3. Dans le fichier compressé, écrire le codage utilisé.
4. Enfin, relire le fichier original, et pour chacun de ses caractères, écrire son code dans le fichier compressé.

Difficulté : la plus petite quantité qu'on peut écrire : **1 octet**.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\quad a \quad}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{a}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot a

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\quad a \quad b \quad c \quad d \quad r}{C = \{000, 010, 011, 10, 11\}}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot ab

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\quad a \quad b \quad c \quad d \quad r}{C = \{000, 010, 011, 10, 11\}}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010**11**0000**11**000**1**0000**1**011000 code le mot abr

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{a}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot abra

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{a}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot abrac

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{000}, \underset{\substack{\\ b}}{010}, \underset{\substack{\\ c}}{011}, \underset{\substack{\\ d}}{10}, \underset{\substack{\\ r}}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 00001011000011000100000101100 0 code le mot abraca

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 00001011000011000100000011000 code le mot abracad

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot abracada

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 00001011000011000100000101 1000 code le mot abracadab

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 00001011000011000100000101 11000 code le mot abracadabr

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de **01**001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{a}{000}, \underset{b}{010}, \underset{c}{011}, \underset{d}{10}, \underset{r}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot abracadabra

Codes préfixes

- Appelons **mot** une suite de **0** et de **1**. Par exemple **01001**.
- Un mot x est **préfixe** d'un mot y si y commence par x.

  Par exemple, **01** est préfixe de 01001.
- Un ensemble fini de mots C est un **code préfixe** si aucun mot de C n'est préfixe d'un autre mot de C.
- Exemple : $\underset{\substack{\\ a}}{C} = \{\underset{\substack{\\ a}}{000}, \underset{\substack{\\ b}}{010}, \underset{\substack{\\ c}}{011}, \underset{\substack{\\ d}}{10}, \underset{\substack{\\ r}}{11}\}$.
- Un mot ne peut pas se décomposer de 2 façons différentes.
- On décode en lisant la suite de 0 et de 1 **de gauche à droite**.
- **Exemple** : la suite 000010110000110001000001011000 code le mot abracadabra

  30 bits au lieu de $11 \times 8 = 88$ bits. **Peut-on faire mieux** ?

Codes préfixes et arbres binaires

- Un **code préfixe** peut être représenté par un **arbre binaire**.
- Chaque feuille correspond à un mot du code : on part de la racine jusqu'à la feuille, en écrivant **0** à chaque fois qu'on descend à gauche et **1** à chaque fois qu'on descend à droite.
- Chaque feuille correspond à un caractère du fichier à coder.
- Exemple : $C = \{000\,, 010\,, 011\,, 10\,, 11\}$.

Codage de Huffman : exemple

On veut coder le texte **MISSISSIPPI MAP**

Difficultés du codage de Huffman

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.
- $\ell_1, \ell_2, \ldots, \ell_n =$ longueurs des codes de $s_1, \ldots, s_n$.

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.
- $\ell_1, \ell_2, \ldots, \ell_n =$ longueurs des codes de $s_1, \ldots, s_n$.
- Espérance de la longueur d'un code :

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.
- $\ell_1, \ell_2, \ldots, \ell_n =$ longueurs des codes de $s_1, \ldots, s_n$.
- Espérance de la longueur d'un code :

$$L = \sum_{k=1}^{n} f_i \ell_i$$

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.
- $\ell_1, \ell_2, \ldots, \ell_n =$ longueurs des codes de $s_1, \ldots, s_n$.
- Espérance de la longueur d'un code :

$$L = \sum_{k=1}^{n} f_i \ell_i$$

Quelle est la longueur du texte produit ?

Codes optimaux

Le codage de Huffman est **optimal** au sens suivant :

- Symboles originaux : $s_1, \ldots, s_n$.
- $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$: fréquences correspondantes.
- $\ell_1, \ell_2, \ldots, \ell_n =$ longueurs des codes de $s_1, \ldots, s_n$.
- Espérance de la longueur d'un code :

$$L = \sum_{k=1}^{n} f_i \ell_i$$

Quelle est la longueur du texte produit ?

Le codage de Huffman produit **l'espérance minimale**.

Optimalité du codage de Huffman

Plan

Codage LZ78

Compression par dictionnaire :

- On représente par un code les sous-chaînes du texte.

Codage LZ78

Compression par dictionnaire :

- On représente par un code les sous-chaînes du texte.
- Historique : LZ77, LZ78, LZW.

Codage LZ78

Compression par dictionnaire :

- On représente par un code les sous-chaînes du texte.
- Historique : LZ77, LZ78, LZW.
- Développé par Lempel et Ziv.

Codage LZ78

Compression par dictionnaire :

- On représente par un code les sous-chaînes du texte.
- Historique : LZ77, LZ78, LZW.
- Développé par Lempel et Ziv.
- Amélioré (et breveté) par Welsh.

Codage LZ78

Compression par dictionnaire :

- On représente par un code les sous-chaînes du texte.
- Historique : LZ77, LZ78, LZW.
- Développé par Lempel et Ziv.
- Amélioré (et breveté) par Welsh.
- gzip, zip, GIF, TIFF,...

Codage LZ78 : principe et exemple

000010110000110001000001011000

Taux de compression minimal et maximal

- Une chaîne qui se compresse bien ?

Taux de compression minimal et maximal

- Une chaîne qui se compresse bien ?
- Taux de compression ?

Taux de compression minimal et maximal

- Une chaîne qui se compresse bien ?
- Taux de compression ?
- Une chaîne qui se compresse mal ?

Taux de compression minimal et maximal

- Une chaîne qui se compresse bien ?
- Taux de compression ?
- Une chaîne qui se compresse mal ?
- Taux de compression ?

The end.
Bonne continuation !