Codage binaire et compression

Journée des mathématiques

Lycée Janot

Codage binaire : le principe

Caractère

Codage binaire : le principe

Caractère

↓

Nombre décimal entre 0 et 127

Codage binaire : le principe

Caractère

↓

Nombre décimal entre 0 et 127

↓

Nombre binaire entre 0 et 1111111

Codage binaire : le principe

Caractère

$\downarrow$

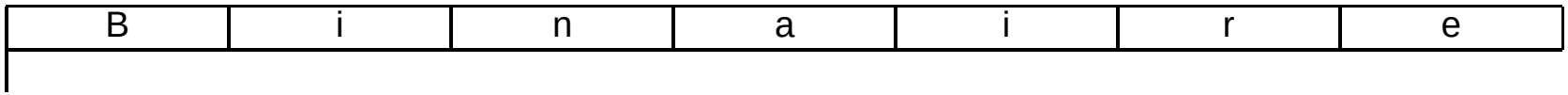
Nombre décimal entre 0 et 127

$\downarrow$

Nombre binaire entre 0 et 1111111

Caractère $\Leftrightarrow$ 7 bits $\leftrightsquigarrow$ 1 octet

Codage binaire : exemple



Codage binaire : exemple

B	i	n	a	i	r	e
66	105	110	97	105	114	101

Codage binaire : exemple

B	i	n	a	i	r	e
66	105	110	97	105	114	101
01000010	01101001	01101110	01100001	01101001	01110010	01100101

Codage binaire : exemple

B	i	n	a	i	r	e
66	105	110	97	105	114	101
01000010	01101001	01101110	01100001	01101001	01110010	01100101

donc :

Binaire $\Leftrightarrow$ 01000010011010010110111001100001011010010111001001100101

Compression RLE : le principe

- run-length encoding.

Compression RLE : le principe

- run-length encoding.
- Toute suite de bits ou de caractères identiques est remplacée par un couple (nombre d'occurrences ; bit ou caractère répété).

Compression RLE : le principe

- run-length encoding.
- Toute suite de bits ou de caractères identiques est remplacée par un couple (nombre d'occurrences ; bit ou caractère répété).
- Le résultat comporte en général moins de caractères, bien que ce ne soit pas une obligation.

Compression RLE : le principe

- run-length encoding.
- Toute suite de bits ou de caractères identiques est remplacée par un couple (nombre d'occurrences ; bit ou caractère répété).
- Le résultat comporte en général moins de caractères, bien que ce ne soit pas une obligation.
  - AAAAAAAAZZEEEEEER donne : 8A2Z6E1R, ce qui est beaucoup plus court.

Compression RLE : le principe

- run-length encoding.
- Toute suite de bits ou de caractères identiques est remplacée par un couple (nombre d'occurrences ; bit ou caractère répété).
- Le résultat comporte en général moins de caractères, bien que ce ne soit pas une obligation.
  - AAAAAAAAZZEEEEEER donne : 8A2Z6E1R, ce qui est beaucoup plus court.
  - WBWBWBWBWB donne : 1W1B1W1B1W1B1W1B1W1B ce qui est deux fois plus long.

Compression RLE : exemple

Binaire

Compression RLE : exemple

Binaire

01000010011010010110111001100001011010010111001001100101

Compression RLE : exemple

Binaire

↓

0100001001101001011011100110000101101001011100100110010 1

↓

10-11-40-11-20-21-10-11-20-11-10-20-10-31-20-21-40-11-10-21-10-11-20-31-20-21-20-11-10-11

Compression RLE : exemple

Binaire

↓

0100001001101001011011100110000101101001011100100110010 1

↓

10-11-40-11-20-21-10-11-20-11-10-20-10-31-20-21-40-11-10-21-10-11-20-31-20-21-20-11-10-11

↓

Compression Huffman : le principe

- Idée : coder ce qui est fréquent sur peu de place, et coder en revanche sur des séquences plus longues ce qui revient rarement.

Compression Huffman : le principe

- Idée : coder ce qui est fréquent sur peu de place, et coder en revanche sur des séquences plus longues ce qui revient rarement.
- Étapes :
  - estimation des fréquences d'apparition des charactères
  - création d'un arbre
  - encodage du texte selon l'arbre

Compression Huffman : le principe

- Idée : coder ce qui est fréquent sur peu de place, et coder en revanche sur des séquences plus longues ce qui revient rarement.
- Étapes :
  - estimation des fréquences d'apparition des charactères
  - création d'un arbre
  - encodage du texte selon l'arbre
- Inconvenients :
  lire tout le fichier avant de comprimer.
  Pour décomprimer il faut connaître les codes et donc la table, qui est ajoutée devant le fichier, aussi bien pour transmettre que stocker, ce qui diminue la compression, surtout pour les petits fichiers.
  Plusieurs variantes de Huffman existent pour supprimer ces defauts.

Exemple

Supposons que la répartition des lettres d'un texte par nombres d'occurrences décroissants soit :

e	a	d	c	b	f	g
100	70	35	30	25	10	5

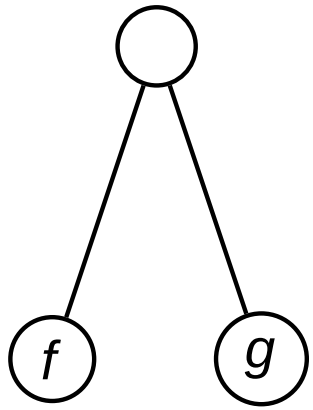
Exemple

Supposons que la répartition des lettres d'un texte par nombres d'occurrences décroissants soit :

e	a	d	c	b	f	g
100	70	35	30	25	10	5

On remplace les deux lettres les moins fréquentes de la liste triée par une lettre (fictive) f+g de nombre d'occurence 10+5=15 que l'on insère, en respectant l'ordre dans la liste triée. On obtient :

e	a	d	c	b	f+g
100	70	35	30	25	15

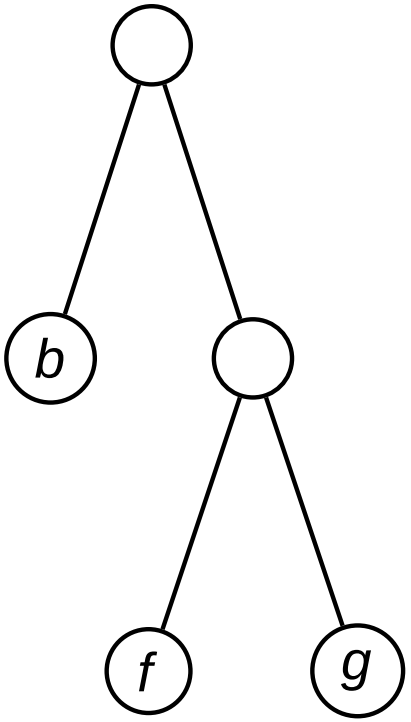


Exemple

e	a	d	c	b	f+g
100	70	35	30	25	15

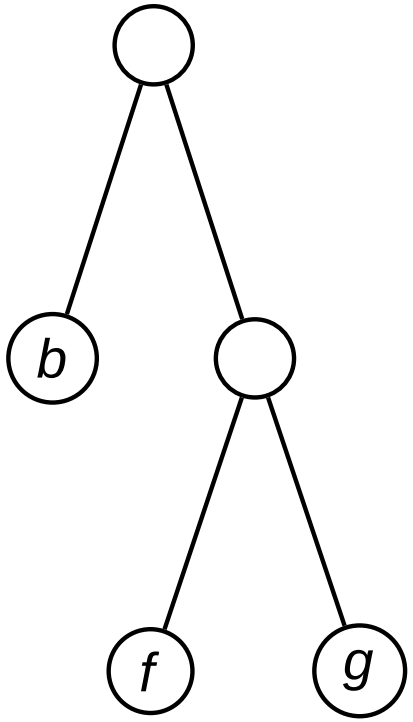
Exemple

e	a	d	c	b	f+g
100	70	35	30	25	15



Exemple

e	a	d	c	b	f+g
100	70	35	30	25	15



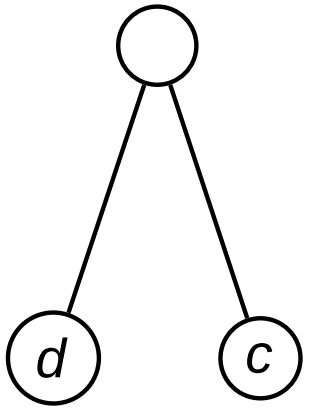
e	a	b+f+g	d	c
100	70	40	35	30

Exemple

e	a	b+f+g	d	c
100	70	40	35	30

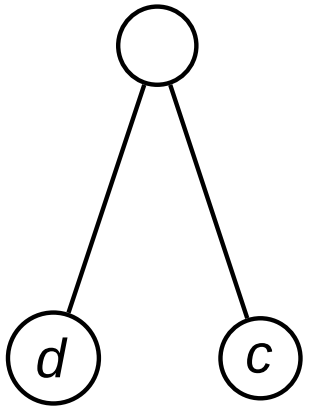
Exemple

e	a	b+f+g	d	c
100	70	40	35	30



Exemple

e	a	b+f+g	d	c
100	70	40	35	30



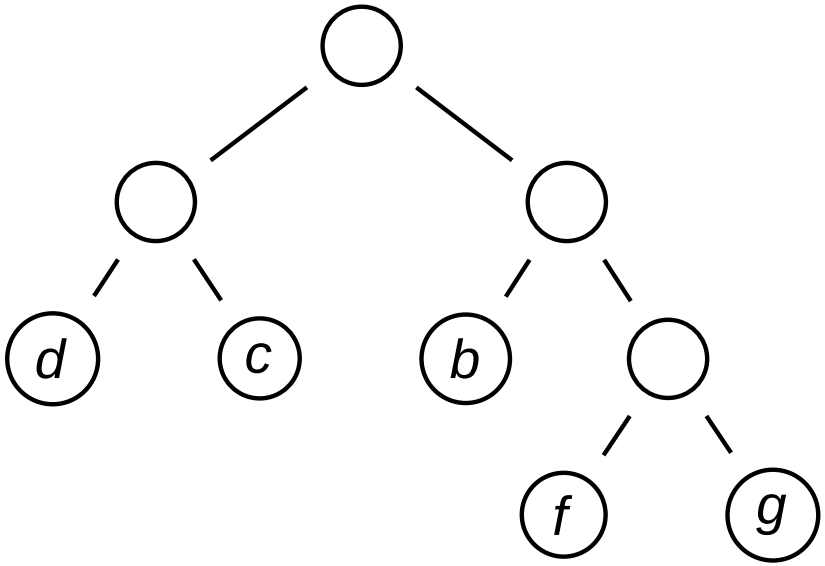
e	a	d+c	b+f+g
100	70	65	40

Exemple

e	a	d+c	b+f+g
100	70	65	40

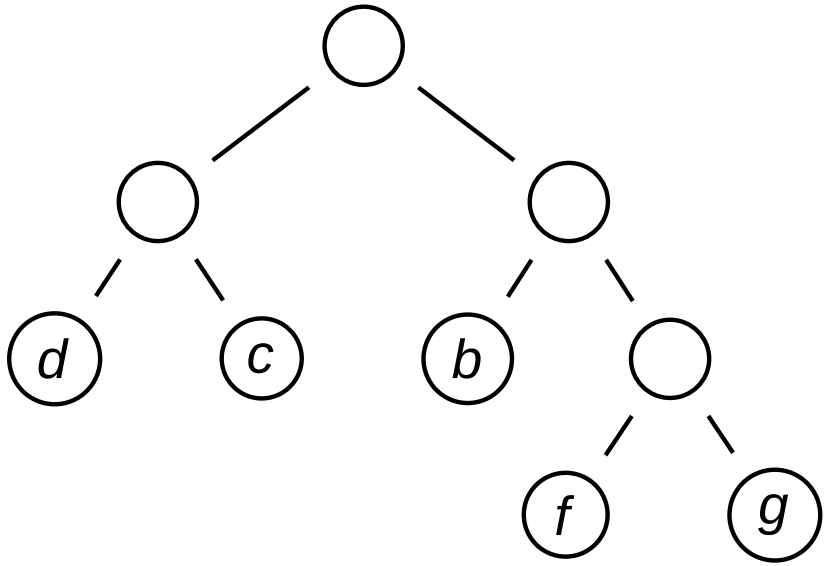
Exemple

e	a	d+c	b+f+g
100	70	65	40



Exemple

e	a	d+c	b+f+g
100	70	65	40



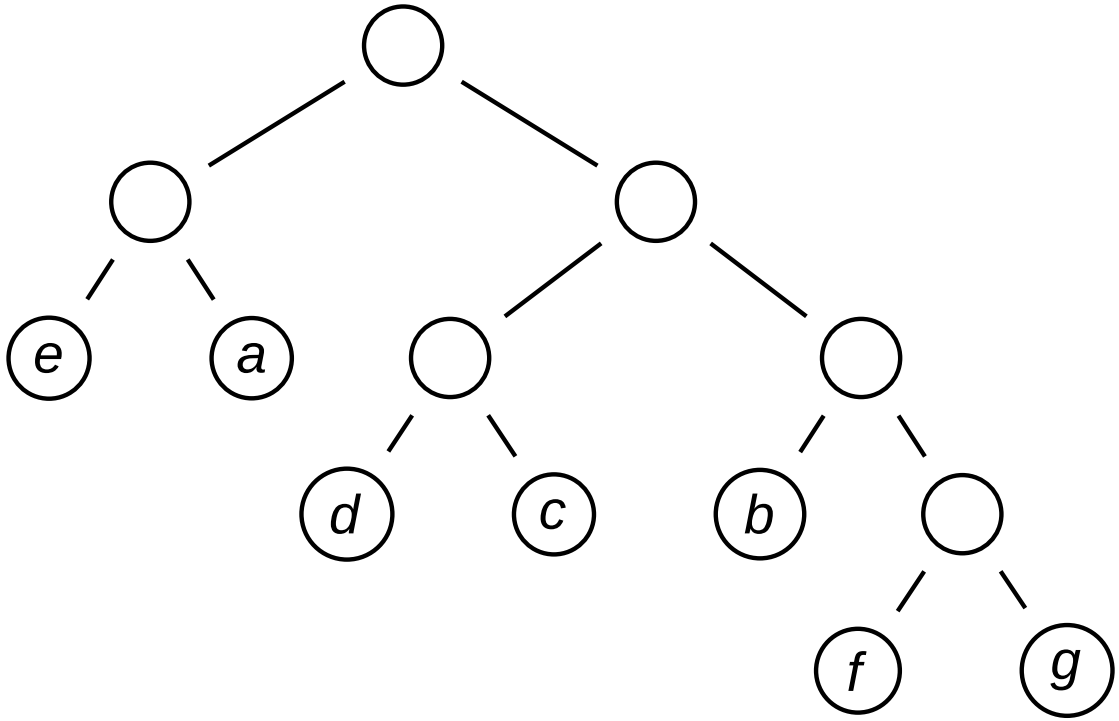
d+c+b+f+g	e	a
105	100	70

Exemple

d+c+b+f+g	e	a
105	100	70

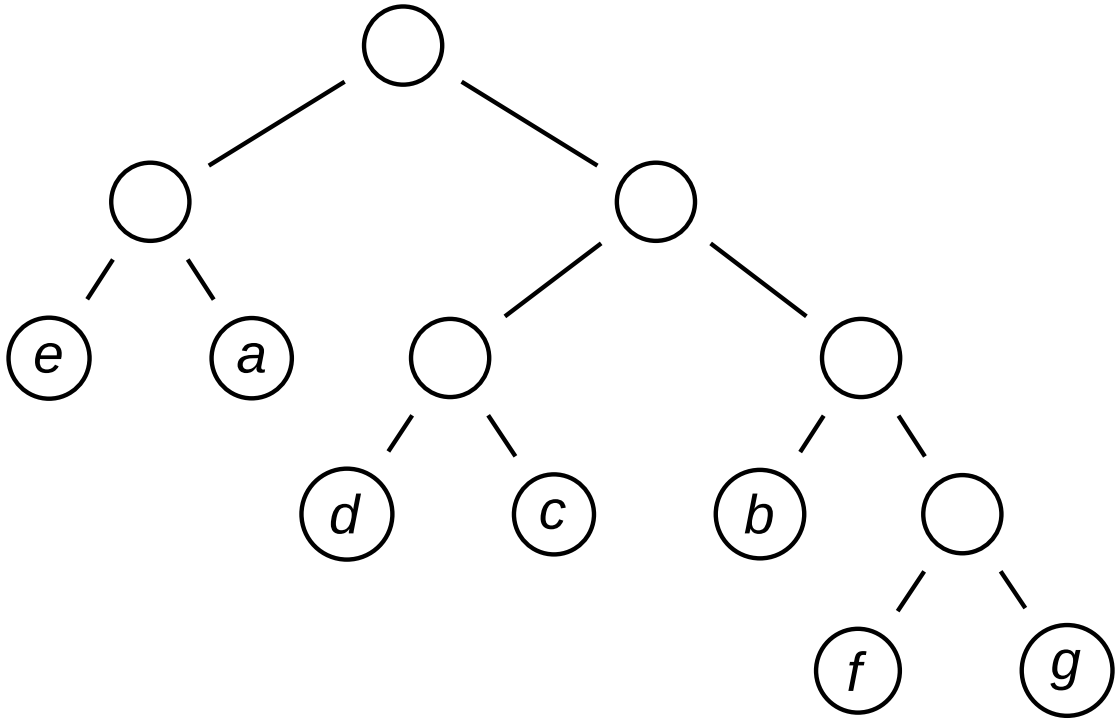
Exemple

d+c+b+f+g	e	a
105	100	70



Exemple

d+c+b+f+g	e	a
105	100	70



e	a	d	c	b	f	g
00	01	100	101	110	1110	1111

Exemple

On obtient :

e	a	d	c	b	f	g
100	70	35	30	25	10	5
00	01	100	101	110	1110	1111

Exemple

On obtient :

e	a	d	c	b	f	g
100	70	35	30	25	10	5
00	01	100	101	110	1110	1111

La longueur du texte vaut

$$(70+100)\times 2+(25+30+35)\times 3+(10+5)\times 4 = 340+270+60 = 670\text{bits}$$

Exemple

On obtient :

e	a	d	c	b	f	g
100	70	35	30	25	10	5
00	01	100	101	110	1110	1111

La longueur du texte vaut

$$(70+100)\times 2+(25+30+35)\times 3+(10+5)\times 4 = 340+270+60 = 670\text{bits}$$

au lieu de

$$(100+70+35+30+25+10+5)\times 8 = 2200\text{bits}$$