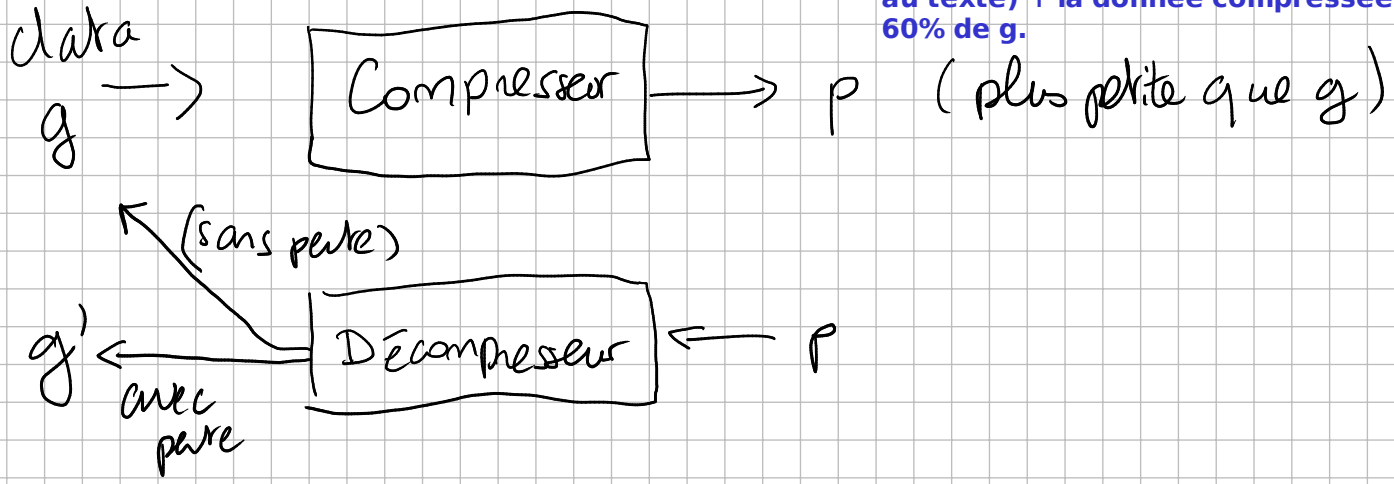
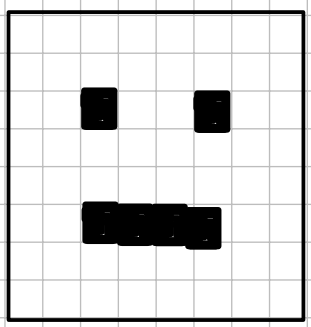
# Compression.




**pour Huffman : l'arbre (petit par rapport au texte) + la donnée compressée 60% de g.**



**8*8 lettres. 0 si c'est blanc et 1 si c'est noir**

**première ligne** 00000000
**seconde ligne** 00000000
**etc** 00100100
00000000
00000000
00111100
00000000
00000000

18 : 0s
1 : 1
2 : 0s
1 : 1
20 : 0s
4 : 1s
18 : 0s

seconde col
0, 1, 0, ...
18, 1, 2, ...
première col

**Une idée de codage. run length encoding.**
**coder les séquences.**

1er chiffre. → 0

liste des distances. → 18, 1, 2, 1, 20, 4, 18

Virgules ?

Comment coder / écrire les longueurs de chaque séquence ?

En binaire.

exercice : écrire ces nombres en binaire.

| | | |
|---|---|---|
| $2^0$ | 1 | 1 |
| $2^1$ | 10 | 2 |
| $2^2$ | 100 | 4 |
| $2^3$ | 1000 | 8 |
| $2^4$ | 10000 | 16 |
| $2^5$ | 100000 | 32 |
| $2^6$ | 1000000 | |

ou + 7 bits écrire sur 8 bits

$18 = 16 + 2 = 10010$ en base 2.
$1 = 2^0 = 1$ — " —
$2 = 2^1 = 10$ — " —
$20 = 16 + 4 = 10100$ — " —

padding

| | |
|---|---|
| 18 | 0 0 0 1 0 0 1 0 |
| 1 | 1 0 0 0 0 0 0 1 |
| 2 | 1 0 0 0 0 0 1 0 |
| 1 | 1 0 0 0 0 0 0 1 |
| 20 | 0 0 0 1 0 1 0 0 |
| 4 | 1 0 0 0 0 1 0 0 |
| 18 | 0 0 0 1 0 0 1 0 |

← une proposition "sans virgule" procède par alignement.

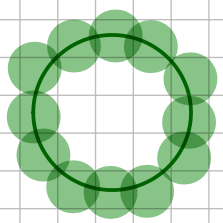
Ou vert un bit de parité

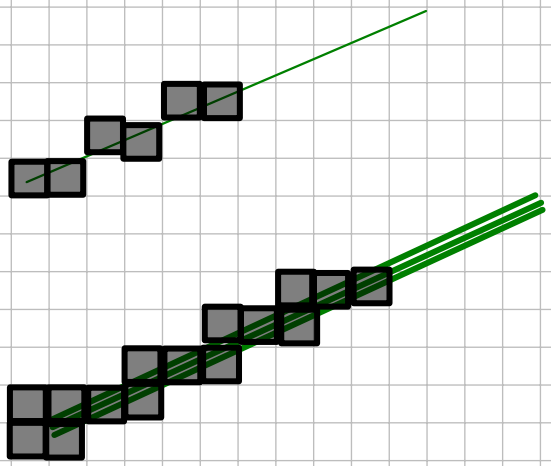
**première ligne** 00000000
**seconde ligne** 00000000
**etc** 00100100
00000000
00000000
00111100
00000000
00000000

00000000

00100100

00111100





**DECODE_MOI**

**Exemple de Huffman**

**message en ASCII 7 bits**
**10*7 = 70 bits.**

**01110011101011100001111001 0**

**une trentaine de bits.**

| | |
|---|---|
| D : 2 | D : 2 |
| E : 2 | E : 2 |
| C : 1 | O : 2 |
| O : 2 | C : 1 |
| _ : 1 | _ : 1 |
| M : 1 | M : 1 |
| I : 1 | I : 1 |

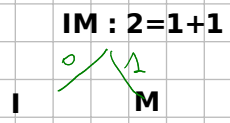
file d'attente de racines d'arbre

lettre — nombre d'occ.

l'algo de Huffman → 1 seul arbre

1.

| |
|---|
| D : 2 |
| E : 2 |
| O : 2 |
| IM:2 |
| C : 1 |
| _ : 1 |

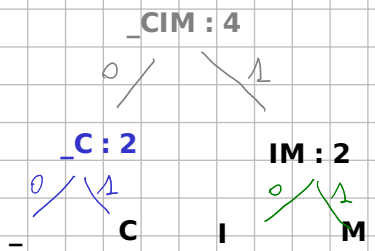



2.

| |
|---|
| D : 2 |
| E : 2 |
| O : 2 |
| IM:2 |
| _C : 2 |

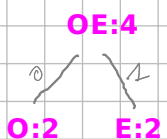

3.

| |
|---|
| _CIM :4 |
| D : 2 |
| E : 2 |
| O : 2 |



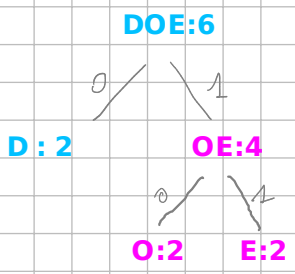

4.

| |
|---|
| _CICM:4 |
| OE : 4 |
| D : 2 |

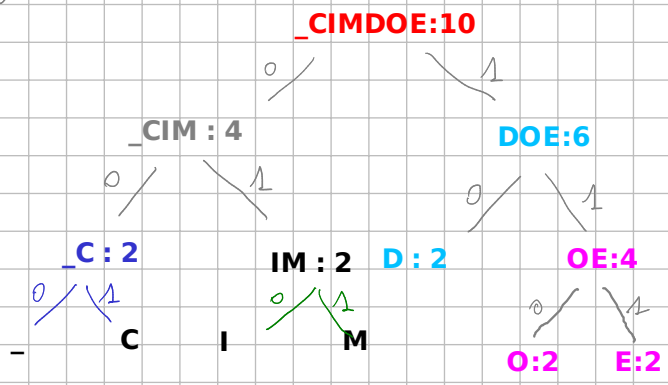


5. -CICM:4

DOE:6




6 _CICMDOE:10

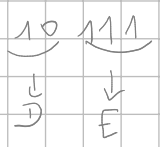



_ : 000
C : 001
I : 010
M : 011
D : 10
O : 110
E : 111

**Compression de Huffman sur le message DECODE_MOI**

**10111001110101110000111110010**

**décoder (sans virgules!) car c'est un code préfixe.**




**Questionnemment : compresser de sorte à permettre de faire des calculs directement sur le compressé?**

**idée qui n'est pas propre à la compression. On la retrouve en cryptographie. chiffrement holomorphique**