Comment en dire autant avec moins de mots ? Les images numériques

Élise Janvresse

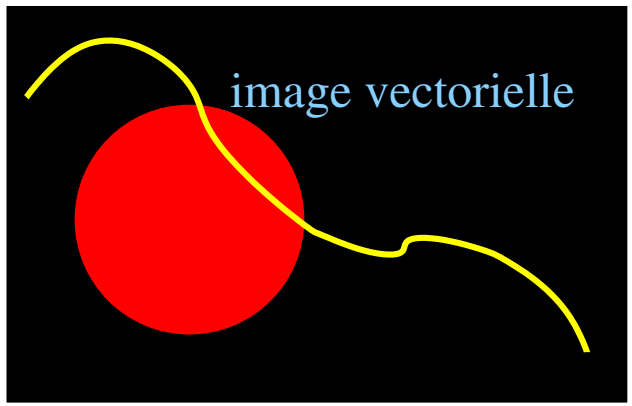
Laboratoire Amiénois de Mathématique Fondamentale et Appliquée

Qu'est-ce qu'une image numérique ?

Qu'est-ce qu'une image numérique ?

Il y en a deux sortes :

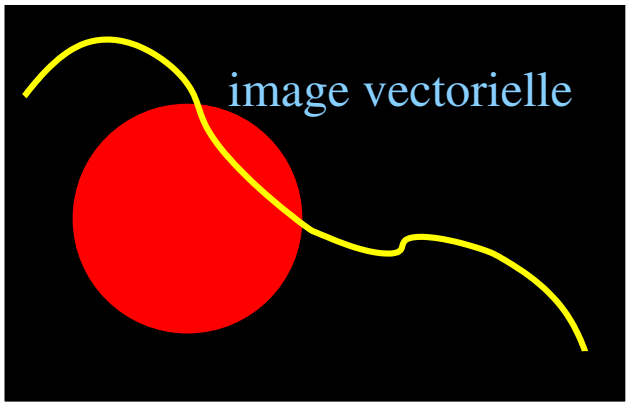
- Les images vectorielles : ensemble d'objets géométriques

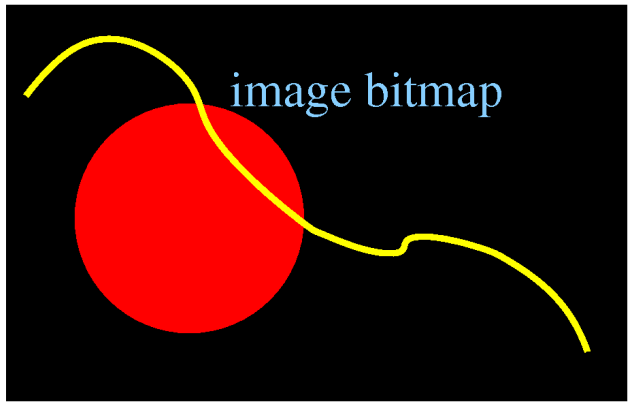


Qu'est-ce qu'une image numérique ?

Il y en a deux sortes :

- Les images vectorielles : ensemble d'objets géométriques





- Les images matricielles (bitmap) : tableau de pixels

Qu'est-ce qu'une image numérique bitmap ?

C'est un ensemble de points appelés pixels (PICture ELement).

Qu'est-ce qu'une image numérique bitmap ?

C'est un ensemble de points appelés pixels (PICture ELement).
Chaque pixel contient une valeur codée en binaire (avec des 0 ou des 1).

Qu'est-ce qu'une image numérique bitmap ?

C'est un ensemble de points appelés pixels (PICture ELement).
Chaque pixel contient une valeur codée en binaire (avec des 0 ou des 1).
* 1 bit/pixel $\rightarrow 2$ couleurs (noir ou blanc).
* 1 octet/pixel $= 8$ bits/pixel $\rightarrow 2^8 = 256$ possibilités.

La couleur

Deux exemples de codage

- RGB ou RVB (Rouge Vert Bleu)

- HSV ou TSV (Teinte Saturation Valeur)

La couleur

Deux exemples de codage

- RGB ou RVB (Rouge Vert Bleu)
  $\rightarrow$ *affecte une valeur (codée sur un octet) à chaque composante de Rouge, de Vert et de Bleu.*

- HSV ou TSV (Teinte Saturation Valeur)

La couleur

Deux exemples de codage

- RGB ou RVB (Rouge Vert Bleu)
  $\rightarrow$ *affecte une valeur (codée sur un octet) à chaque composante de Rouge, de Vert et de Bleu.*
  2^8 *(rouge)* $\times 2^8$ *(vert)* $\times 2^8$ *(bleu)* $= 16\ 777\ 216$ *couleurs.*
- HSV ou TSV (Teinte Saturation Valeur)

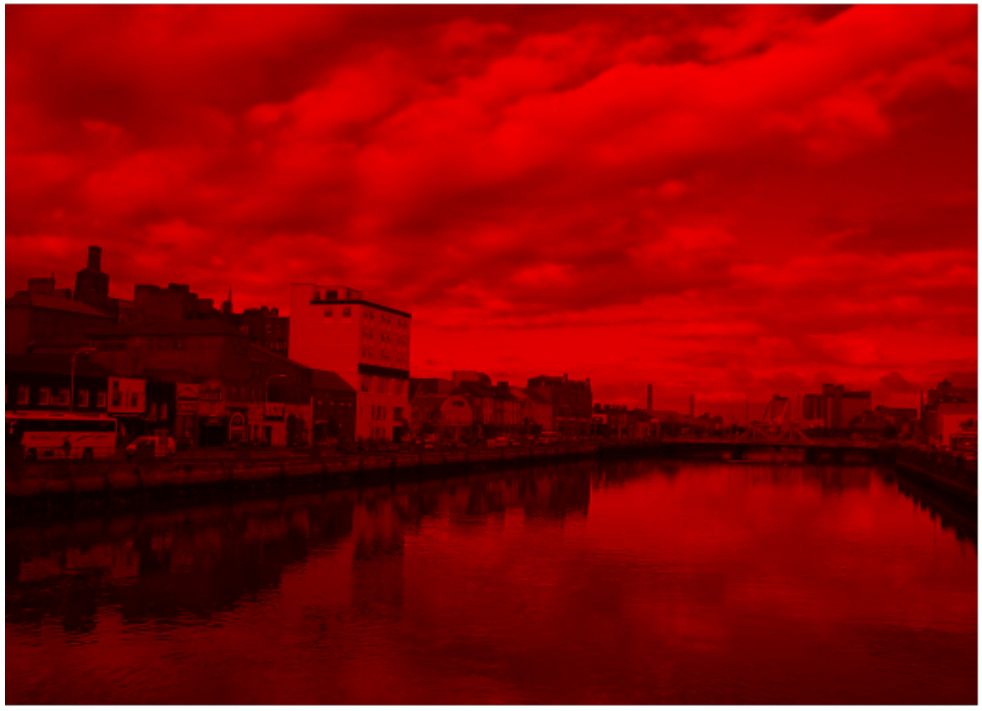
La couleur

Deux exemples de codage

- RGB ou RVB (Rouge Vert Bleu)
  $\rightarrow$ *affecte une valeur (codée sur un octet) à chaque composante de Rouge, de Vert et de Bleu.*
  2^8 *(rouge)* $\times 2^8$ *(vert)* $\times 2^8$ *(bleu)* $= 16\ 777\ 216$ *couleurs.*
- HSV ou TSV (Teinte Saturation Valeur)
  $\rightarrow$ *décompose la couleur selon des critères physiologiques.*

Décomposition en couches RGB

Décomposition en couches RGB



Décomposition en couches RGB

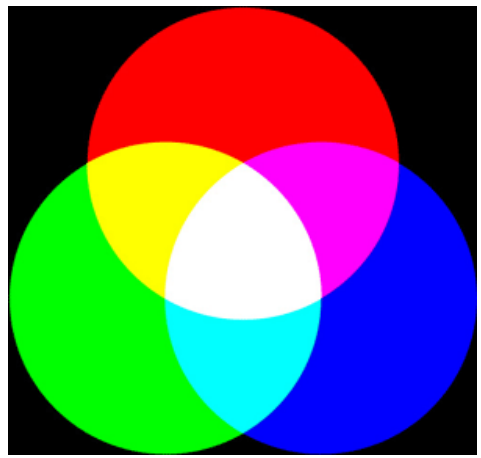
Décomposition en couches RGB

Décomposition en couches RGB

La couleur - Le codage RGB

3 couleurs primaires : Rouge, Vert et Bleu.

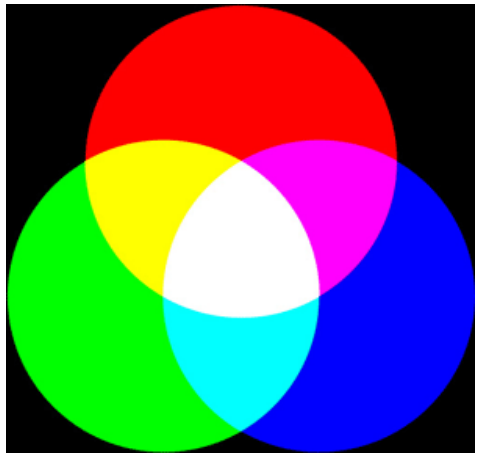
Combinaisons additives



La couleur - Le codage RGB

3 couleurs primaires : Rouge, Vert et Bleu.

Combinaisons additives

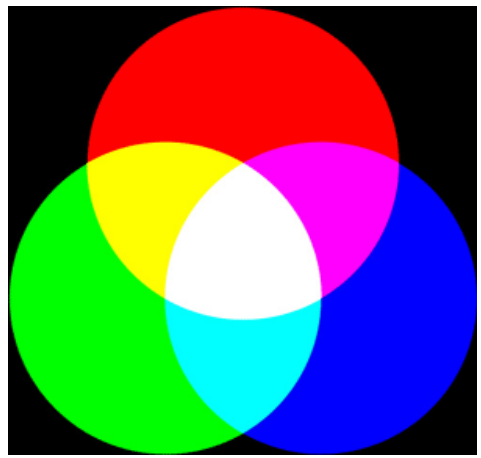


3 couleurs secondaires : cyan, magenta, jaune.

La couleur - Le codage RGB

3 couleurs primaires : Rouge, Vert et Bleu.

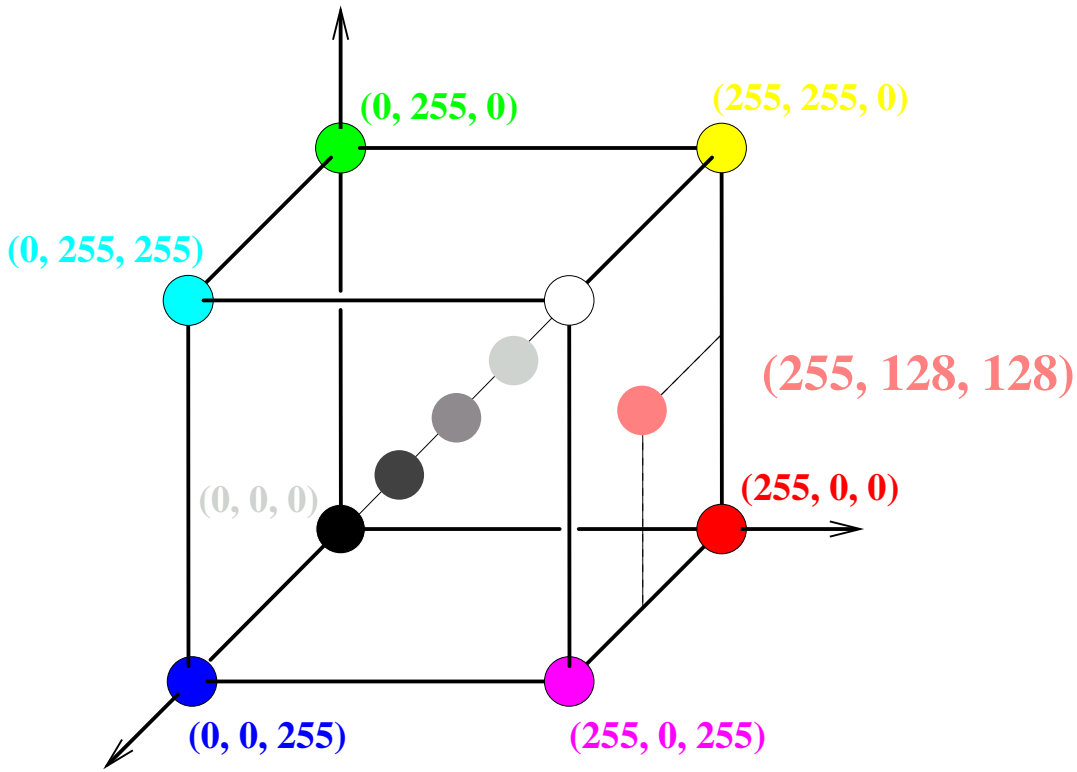
Combinaisons additives



3 couleurs secondaires : cyan, magenta, jaune.
Addition des trois couleurs primaires : blanc.
Absence de composante : noir.

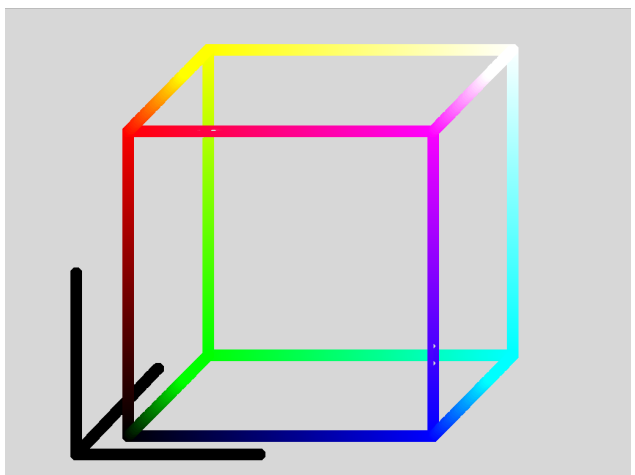
La couleur - Le codage RGB

Chaque couleur correspond à un triplet des quantités de chacune des couleurs primaires.



La couleur

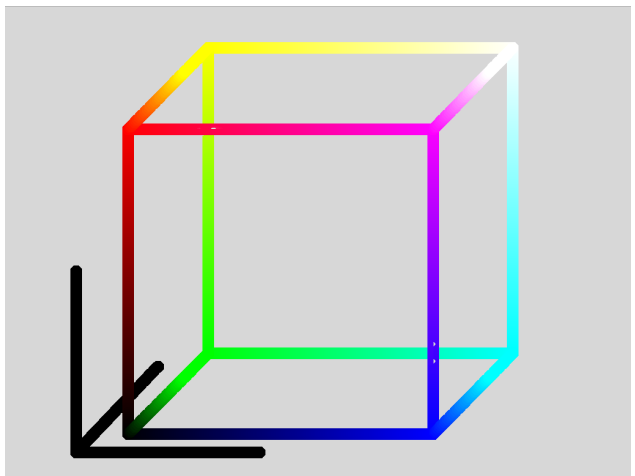
Base : Toute couleur peut être écrite de façon unique comme *combinaison linéaire* des couleurs rouge, vert, bleu.



La couleur

Base : Toute couleur peut être écrite de façon unique comme *combinaison linéaire* des couleurs rouge, vert, bleu.

nouvelle couleur $= 1.$Rouge$+ \frac{128}{255}.$Vert $+ \frac{128}{255}.$Bleu

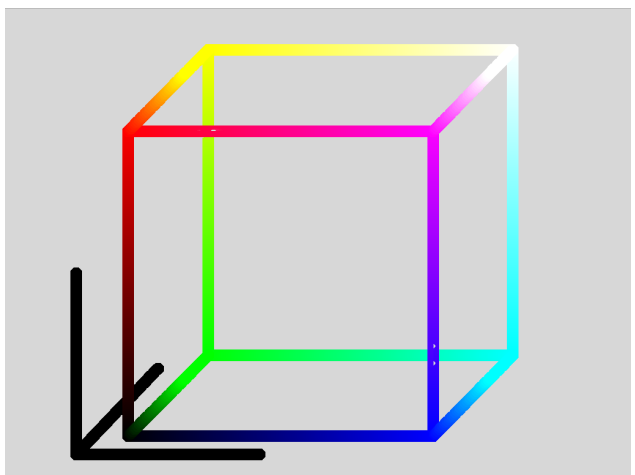


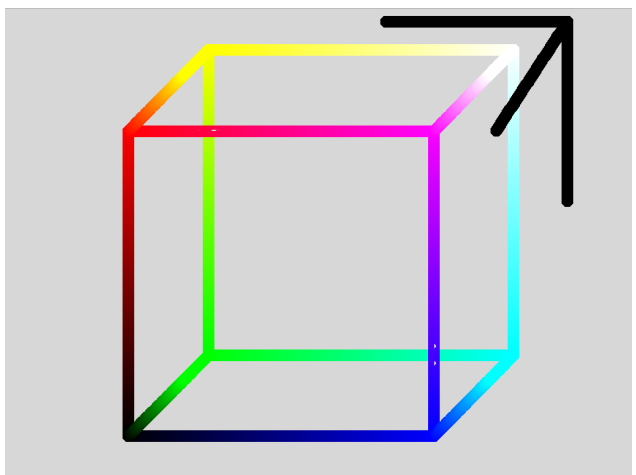
La couleur

Base : Toute couleur peut être écrite de façon unique comme *combinaison linéaire* des couleurs rouge, vert, bleu.

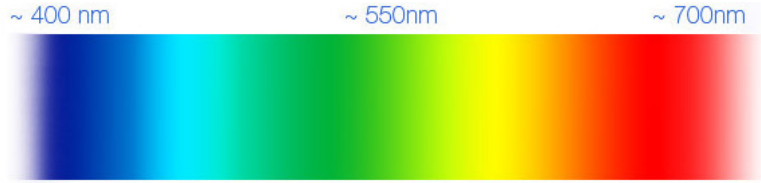
nouvelle couleur $= 1.$Rouge$+ \frac{128}{255}.$Vert $+ \frac{128}{255}.$Bleu

Autre base : On peut aussi écrire toute couleur de façon unique comme *combinaison linéaire* des couleurs cyan, jaune, magenta.



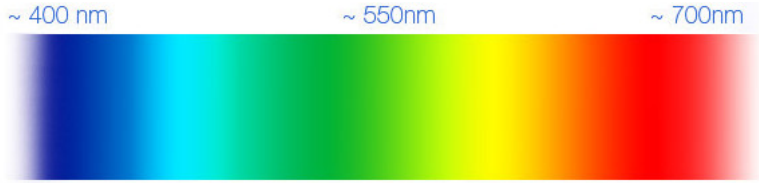


La couleur - Le codage HSV



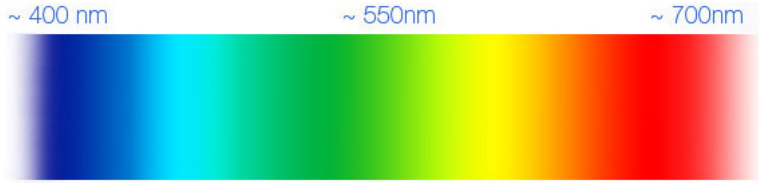
Teinte : couleur de base.

La couleur - Le codage HSV



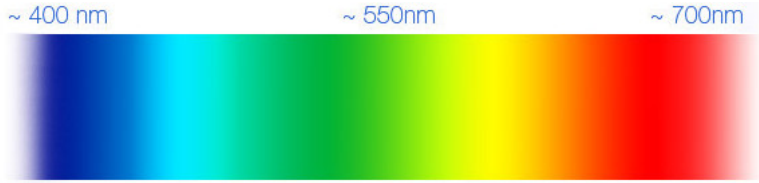
Teinte : couleur de base.
Saturation : caractère vif ou terne de la couleur (pureté).

La couleur - Le codage HSV

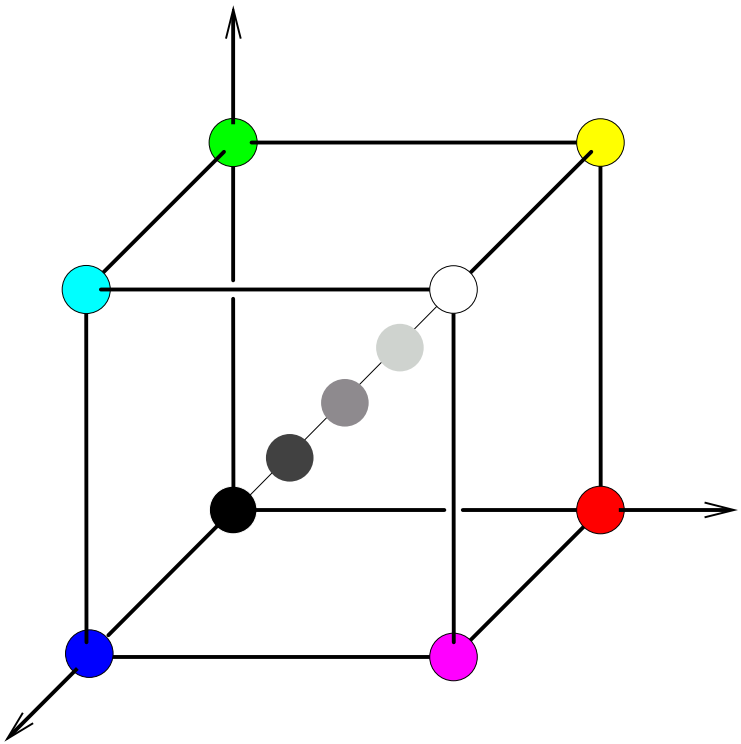


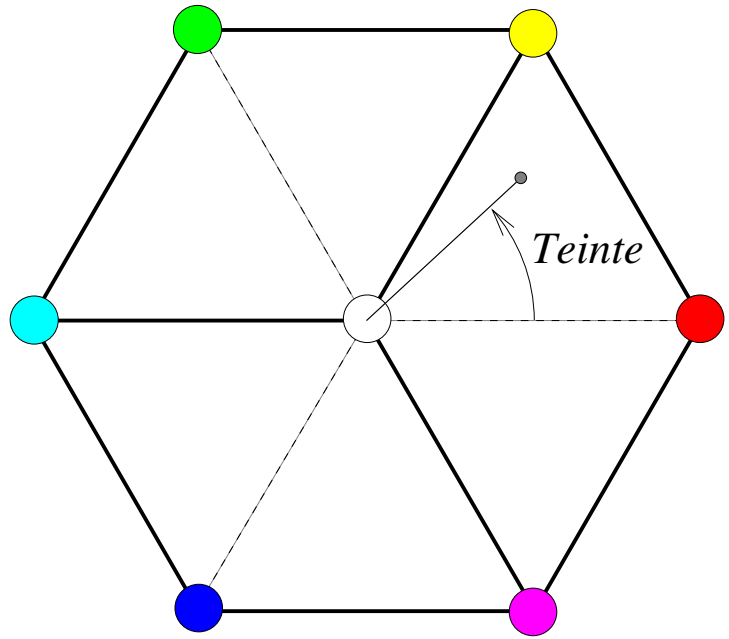
Teinte : couleur de base.
Saturation : caractère vif ou terne de la couleur (pureté).
Valeur : aspect clair ou sombre (brillance).

La couleur - Le codage HSV



Teinte : couleur de base.
Saturation : caractère vif ou terne de la couleur (pureté).
Valeur : aspect clair ou sombre (brillance).





Manipulation des images

- dans le domaine spatial
- dans le domaine des fréquences

Manipulation des images

- dans le domaine spatial
  $\rightarrow$ *manipulation directe des pixels.*
- dans le domaine des fréquences

Manipulation des images

- dans le domaine spatial
  $\rightarrow$ *manipulation directe des pixels.*
- dans le domaine des fréquences
  $\rightarrow$ *transformée de Fourier.*

Domaine spatial

Image en noir et blanc

$\longrightarrow$ matrice dont les coefficients correspondent aux niveaux de gris des pixels (entre 0 et 255)

Domaine spatial

Image en noir et blanc
$\longrightarrow$ matrice dont les coefficients correspondent aux niveaux de gris des pixels (entre 0 et 255)

- Manipulation pixel par pixel

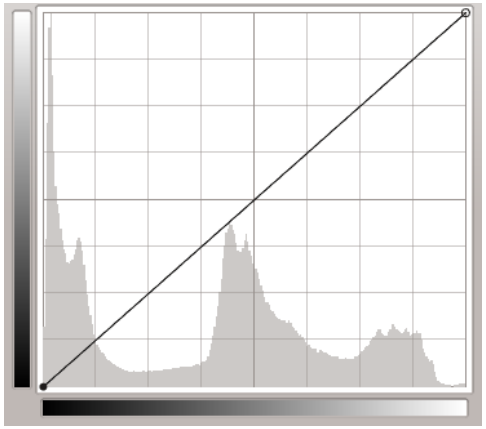
Domaine spatial

Image en noir et blanc
$\longrightarrow$ matrice dont les coefficients correspondent aux niveaux de gris des pixels (entre 0 et 255)

- Manipulation pixel par pixel
  Négatif : $h(x) = 255 - x$

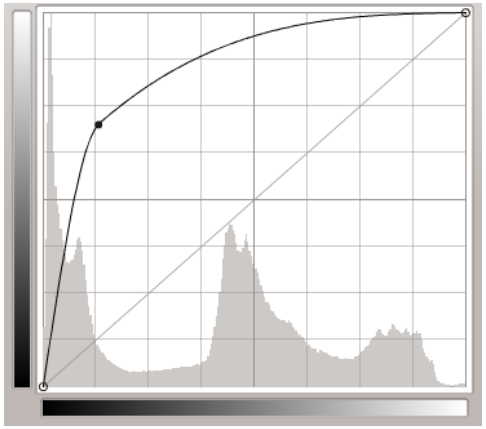
Domaine spatial

- Manipulation pixel par pixel
  changement de niveaux de gris



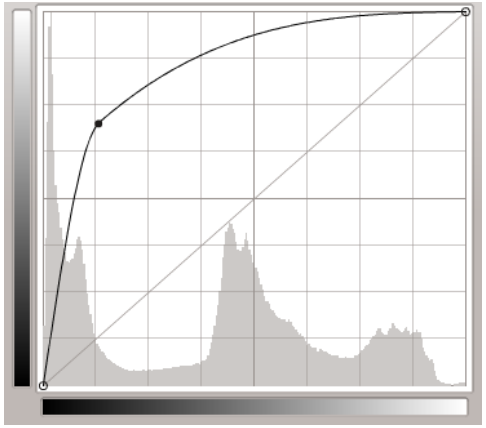
Domaine spatial

- Manipulation pixel par pixel
  changement de niveaux de gris



Domaine spatial

- Manipulation pixel par pixel
  changement de niveaux de gris



Domaine spatial

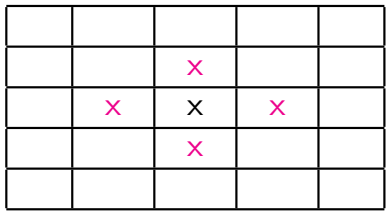
Utilisation des valeurs dans les pixels voisins

convolution

Domaine spatial

Utilisation des valeurs dans les pixels voisins

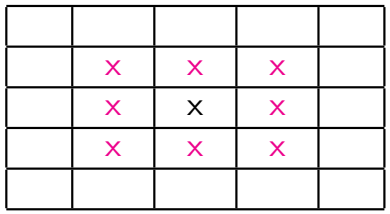
convolution



Domaine spatial

Utilisation des valeurs dans les pixels voisins

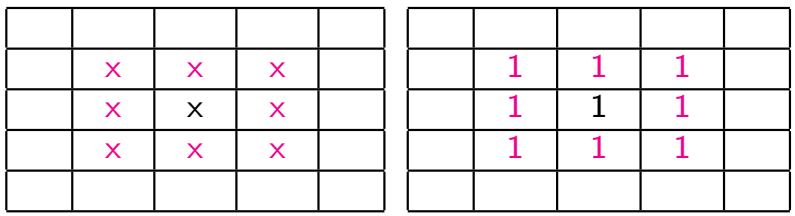
convolution



Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution



$\frac{1}{9} \Longrightarrow$ lissage

Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution

	×	×	×	
	×	×	×	
	×	×	×	

	1	1	1	
	1	1	1	
	1	1	1	

$\frac{1}{9} \Longrightarrow$ lissage

Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution

	×	×	×	
	×	×	×	
	×	×	×	

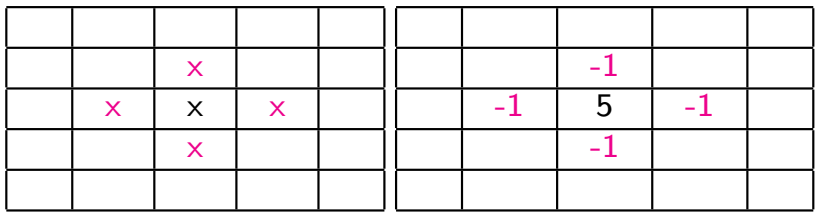
	1	1	1	
	1	1	1	
	1	1	1	

$\frac{1}{9} \Longrightarrow$ lissage

Domaine spatial

Utilisation des valeurs dans les pixels voisins

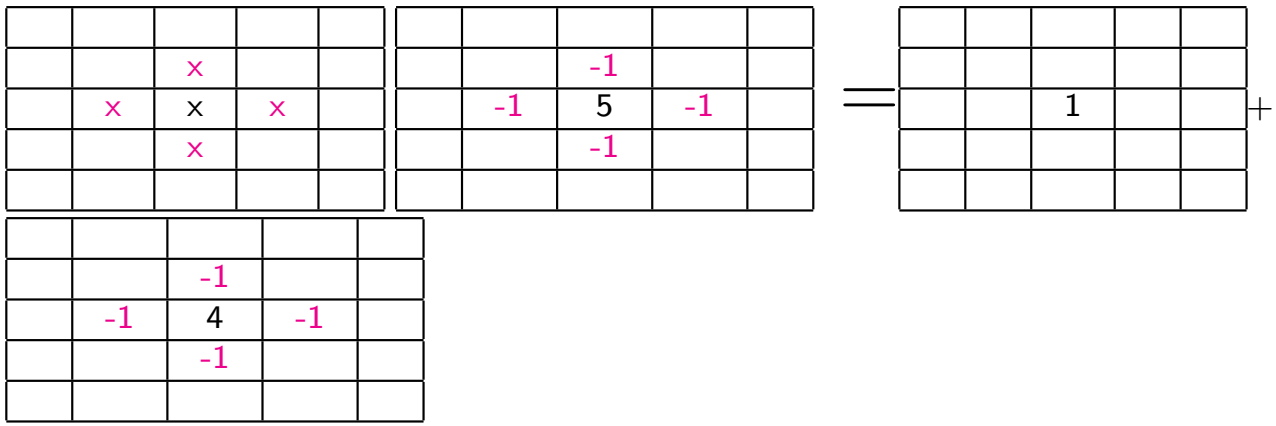
convolution



Domaine spatial

Utilisation des valeurs dans les pixels voisins

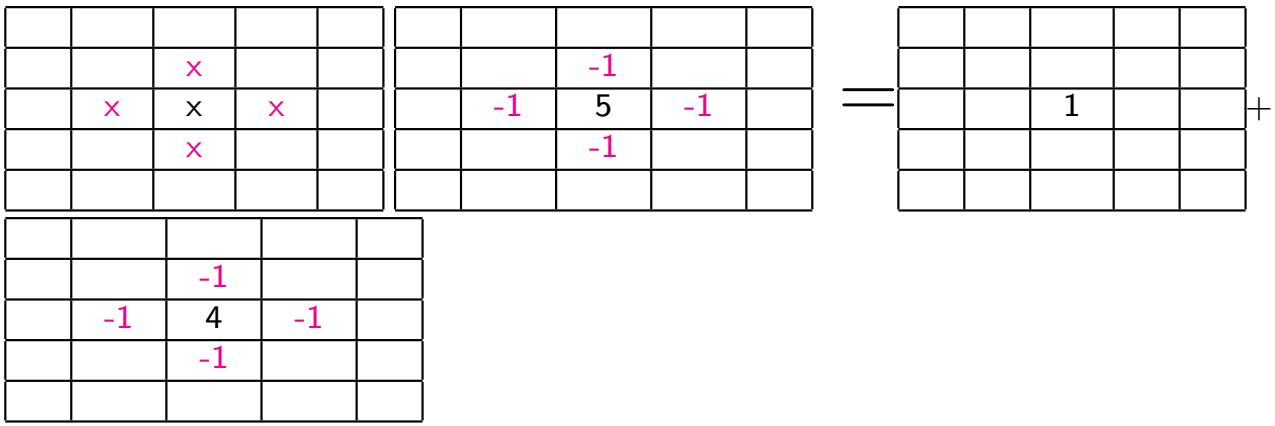
convolution



Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution



$\Longrightarrow$ amplification des bords

Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution

		x		
	x	x	x	
		x		

		-1		
	-1	5	-1	
		-1		

$=$

		1		

$+$

		-1		
	-1	4	-1	
		-1		

$\Longrightarrow$ amplification des bords

Domaine spatial

Utilisation des valeurs dans les pixels voisins

convolution

		x		
	x	x	x	
		x		

		-1		
	-1	5	-1	
		-1		

=

		1		

+

		-1		
	-1	4	-1	
		-1		

$\Longrightarrow$ amplification des bords

Poids d'une image

Poids d'une image

Quantité d'information nécessaire pour coder chaque image : $1\,600 \times 1\,200 \times 3 \simeq 5\,625$ ko.

Poids d'une image

`fichier1.jpg` :
`339 ko`

`fichier2.jpg` :
`995 ko`

Quantité d'information nécessaire pour coder chaque image : $1\,600 \times 1\,200 \times 3 \simeq 5\,625$ ko.

Compression de données

Opération informatique consistant à transformer une suite de bits en une suite de bits plus courte pouvant restituer une information “similaire”

Compression de données

Opération informatique consistant à transformer une suite de bits en une suite de bits plus courte pouvant restituer une information “similaire”

$\longrightarrow$ avec ou sans perte

Codage RLE (run-length encoding)

Principe : Une suite de caractères identiques est remplacée par un couple (nombre d'occurrences ; caractère répété)

Codage RLE (run-length encoding)

Principe : Une suite de caractères identiques est remplacée par un couple (nombre d'occurrences ; caractère répété)

BBBBBBBBNNGGGGGGR donne 8B2N6G1R

Codage RLE (run-length encoding)

Principe : Une suite de caractères identiques est remplacée par un couple (nombre d'occurrences ; caractère répété)

BBBBBBBBNNGGGGGGR donne 8B2N6G1R

Pour une image : toutes les lignes de pixels sont jointes pour former une unique séquence

Codage RLE (run-length encoding)

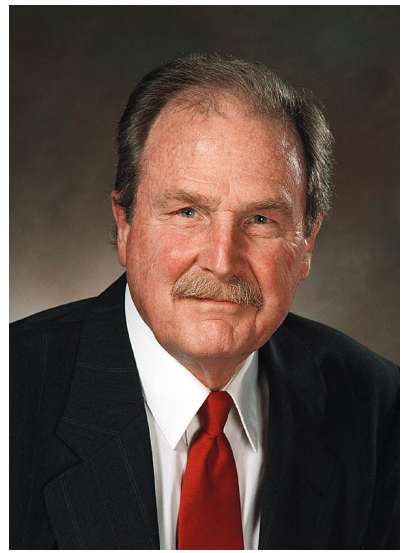
Principe : Une suite de caractères identiques est remplacée par un couple (nombre d'occurrences ; caractère répété)

BBBBBBBBNNGGGGGGR donne 8B2N6G1R

Pour une image : toutes les lignes de pixels sont jointes pour former une unique séquence

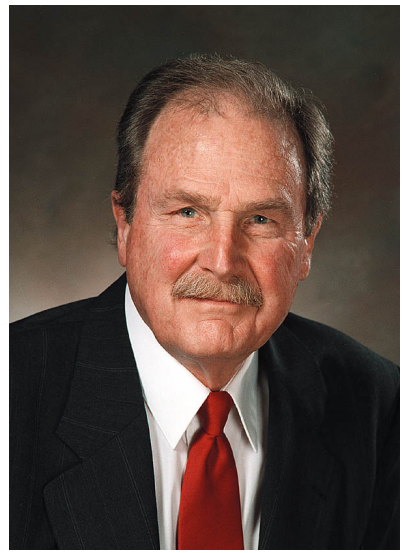
ABABABABABAB donne 1A1B1A1B1A1B1A1B1A1B1A1B

Le codage de Huffman (1952)



David Huffman (1925–1999)

Le codage de Huffman (1952)



David Huffman (1925–1999)

Principe : plus un symbole est utilisé, moins longue doit être la suite de 0 et de 1 qui le code.

Le codage de Huffman

ABRACADABRA

Le codage de Huffman

ABRACADABRA

codage ASCII (7 bits/caractère) : $11 \times 7 \ = \ 77$ bits

Le codage de Huffman

ABRACADABRA

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

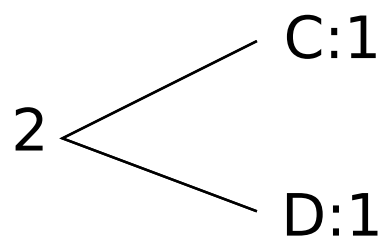
A :5 B :2 R :2 C :1 D :1

Le codage de Huffman

ABRACADABRA

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

A :5 B :2 R :2

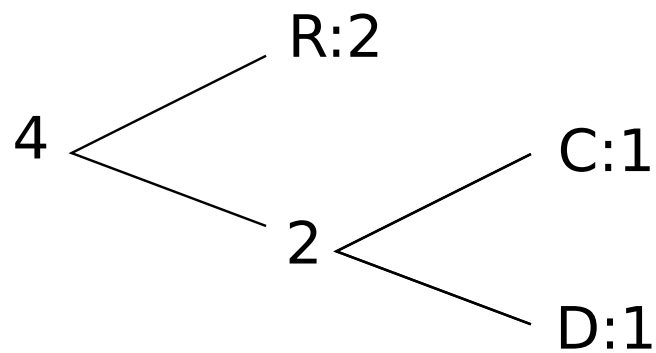


Le codage de Huffman

ABRACADABRA

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

A :5 B :2

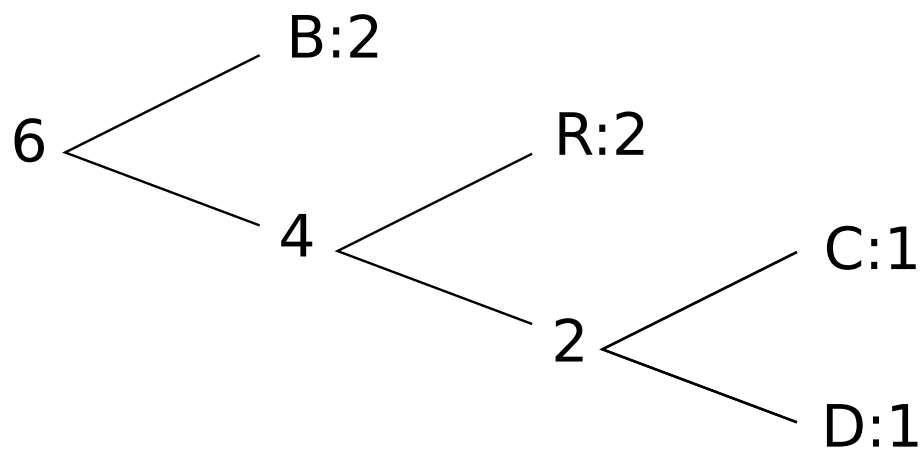


Le codage de Huffman

ABRACADABRA

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

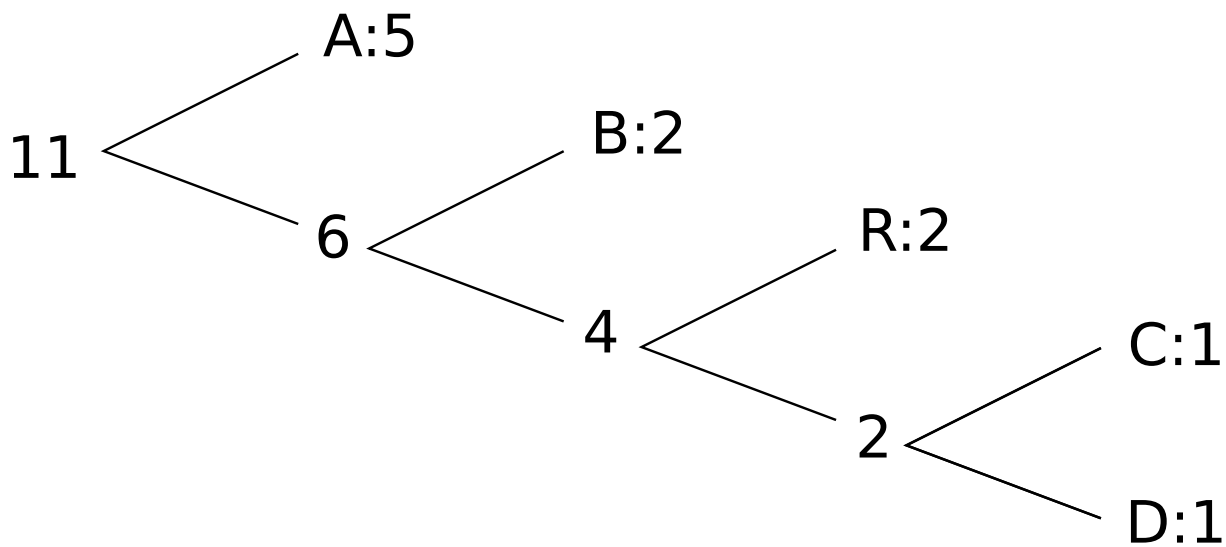
A :5



Le codage de Huffman

ABRACADABRA

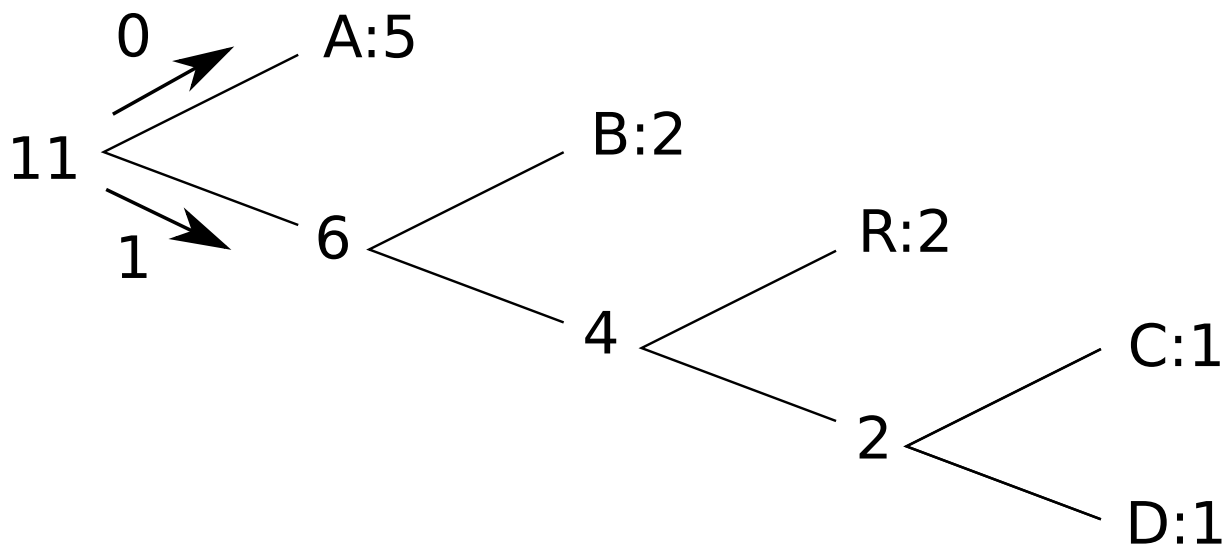
codage ASCII (7 bits/caractère) : $11 \times 7 \ = \ 77$ bits



Le codage de Huffman

ABRACADABRA

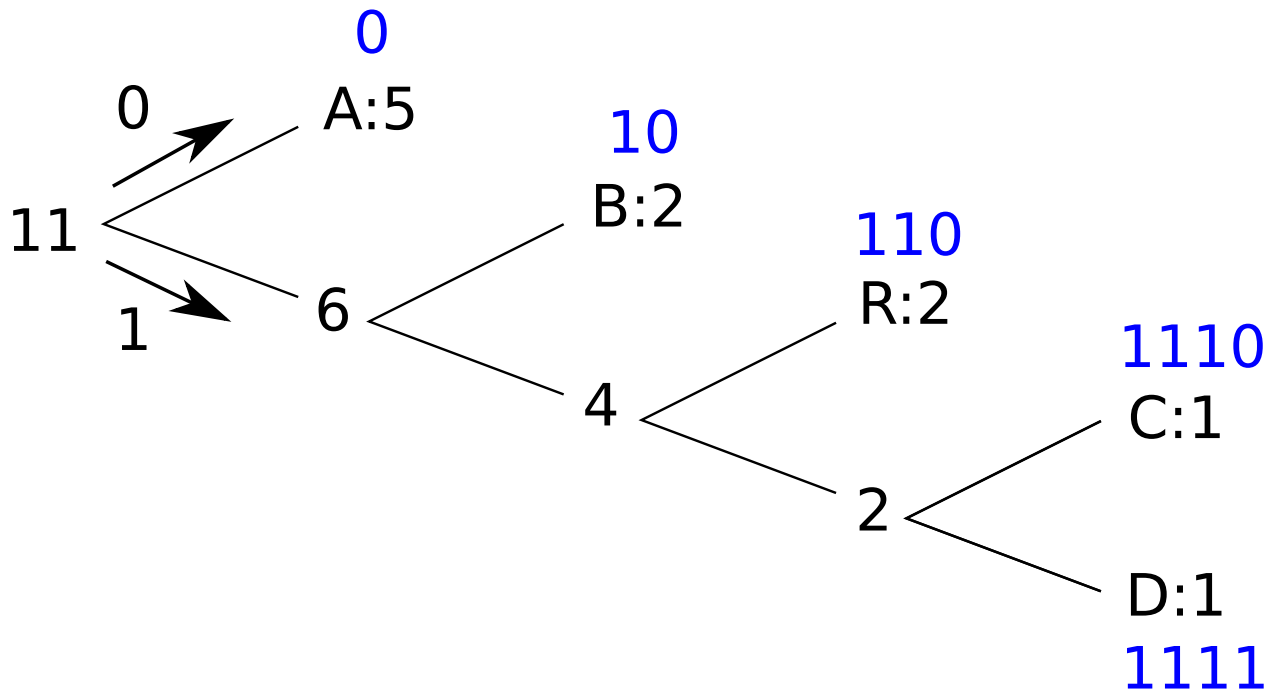
codage ASCII (7 bits/caractère) : $11 \times 7 \ = \ 77$ bits



Le codage de Huffman

ABRACADABRA

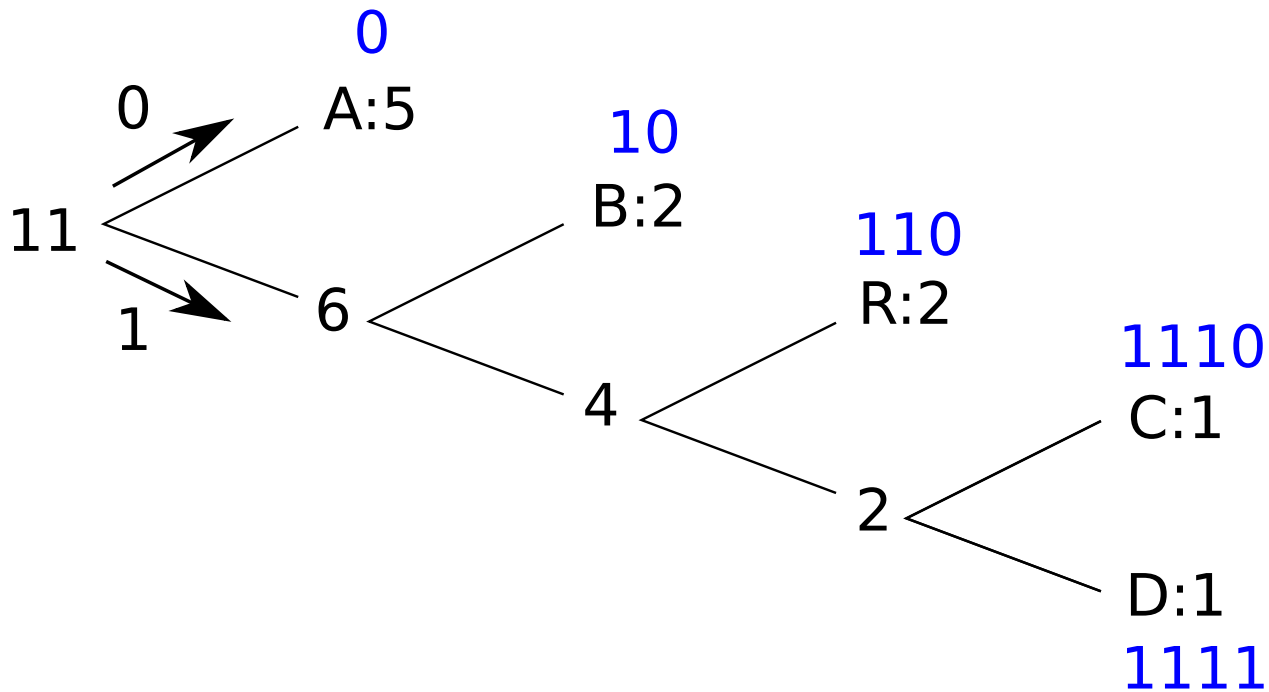
codage ASCII (7 bits/caractère) : $11 \times 7 \ = \ 77$ bits



Le codage de Huffman

ABRACADABRA 0

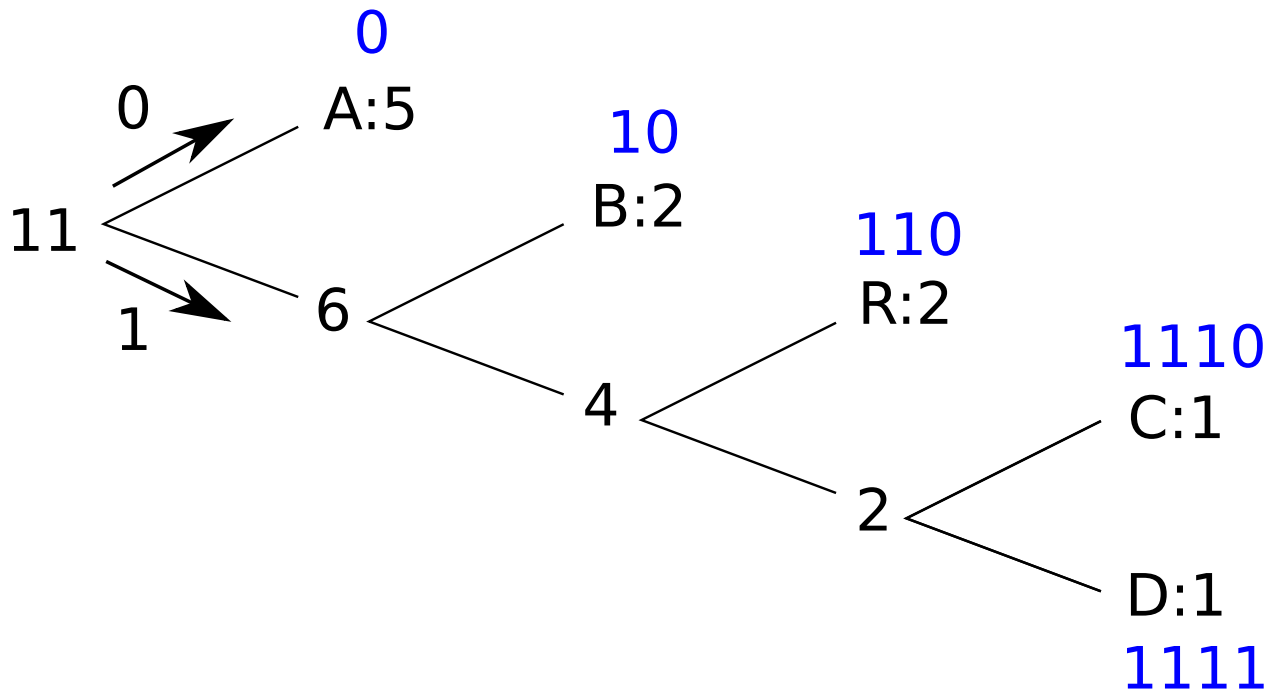
codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits



Le codage de Huffman

ABRACADABRA 010

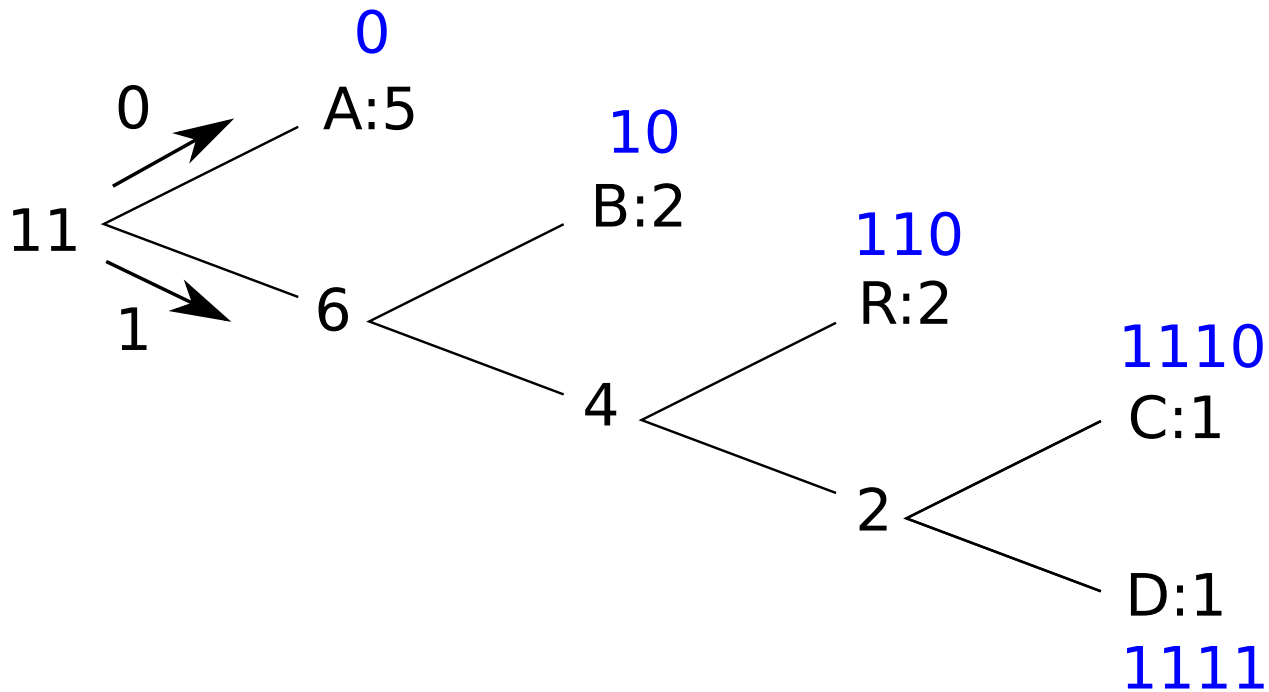
codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits



Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

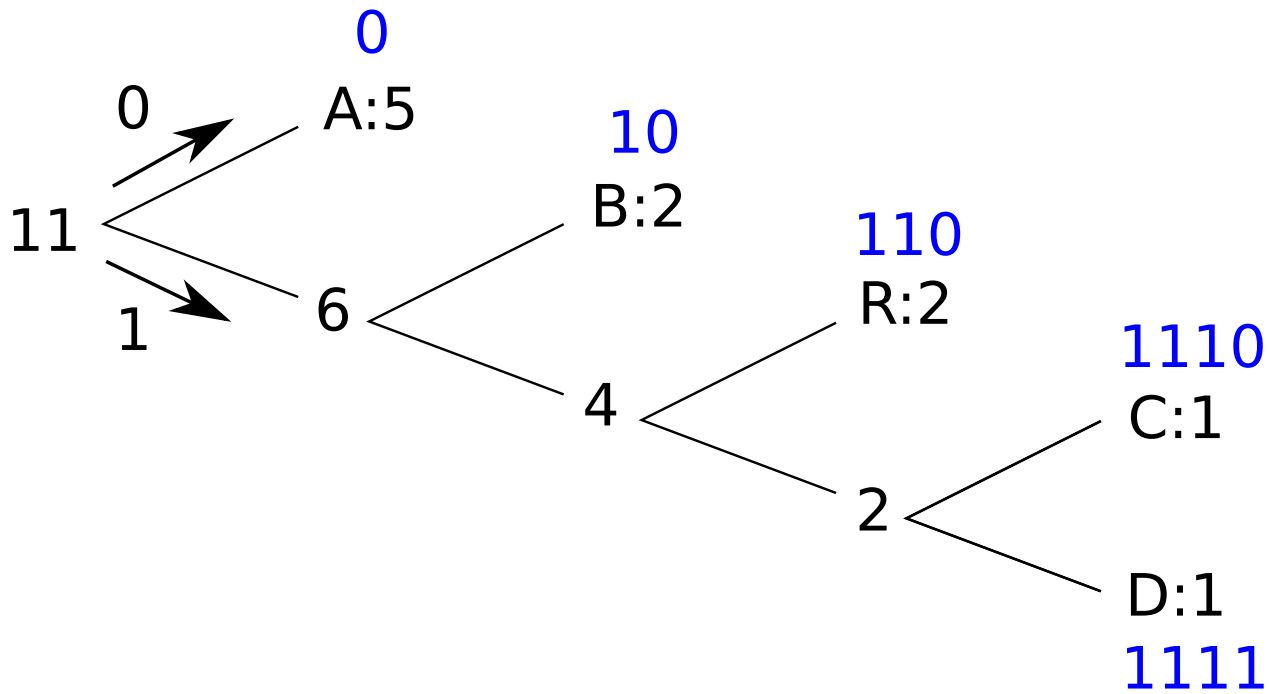


Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !

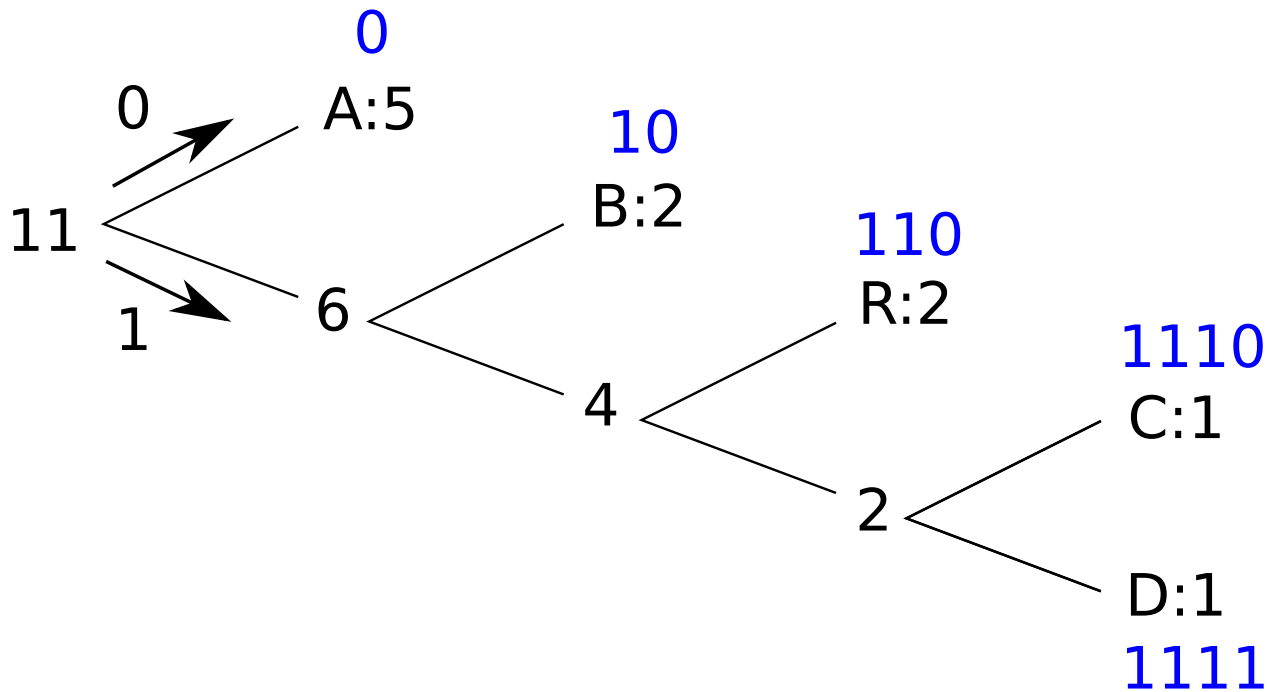


Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !

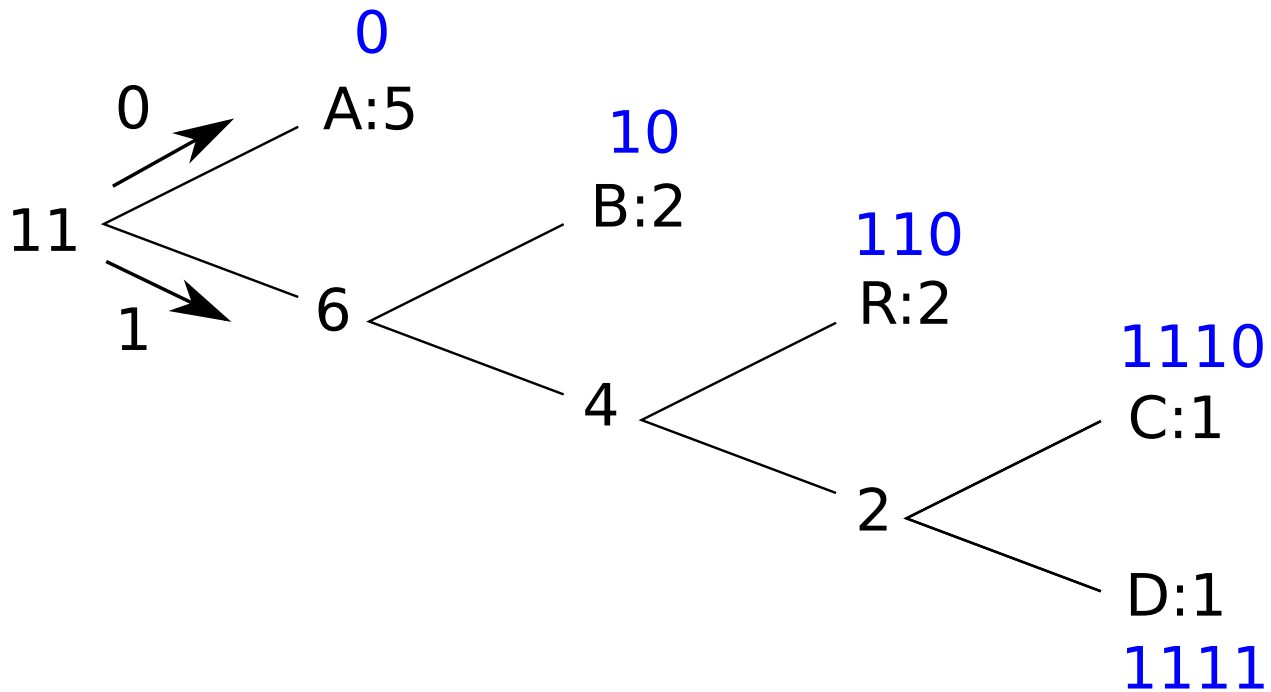


Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !



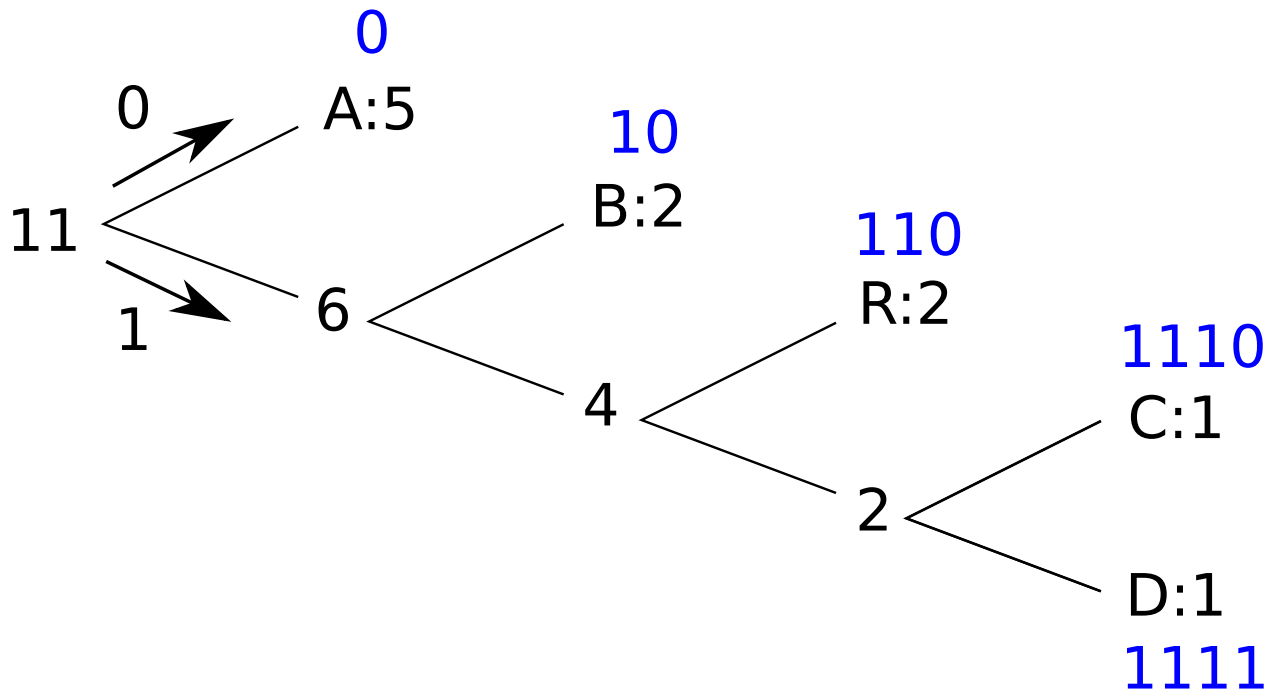
A

Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !



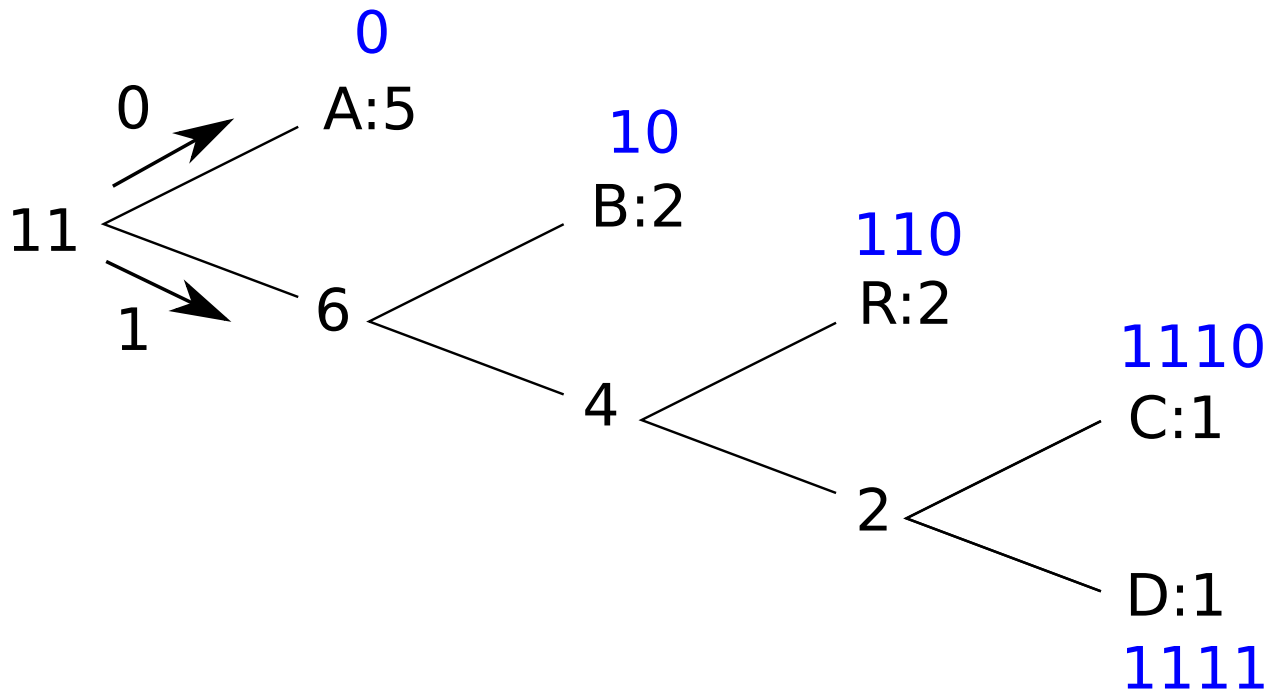
A

Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !



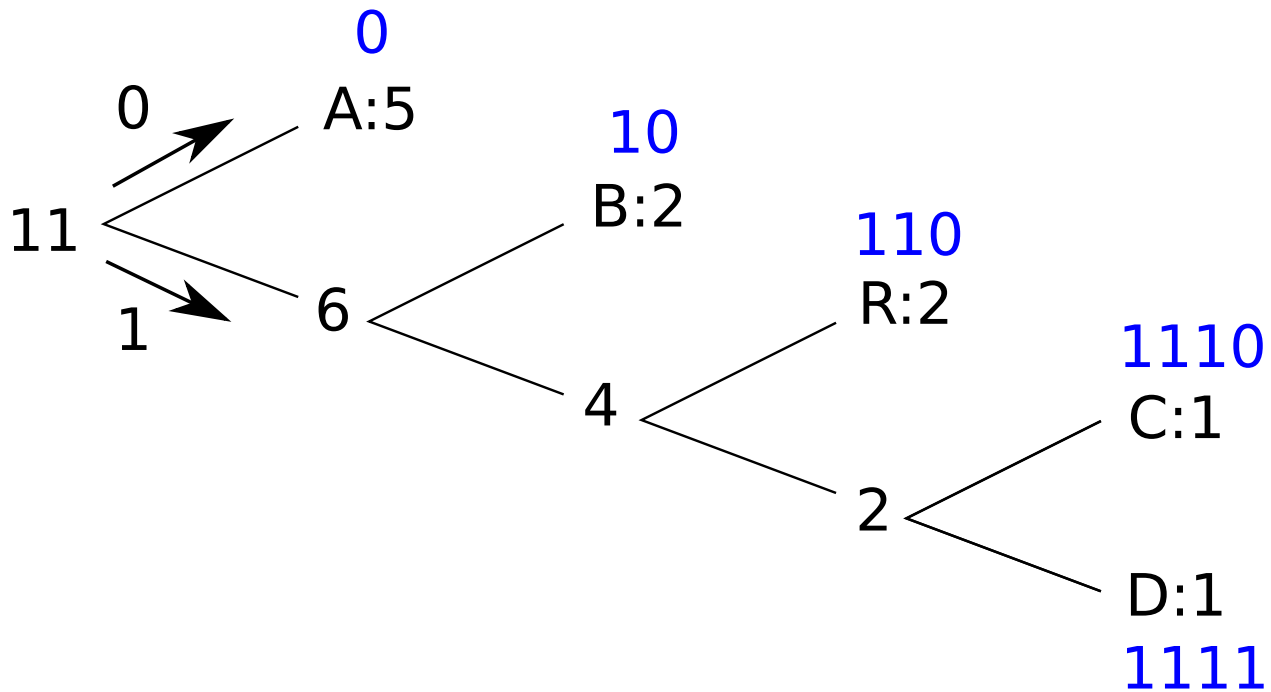
AB

Le codage de Huffman

ABRACADABRA 01011001110011110101100

codage ASCII (7 bits/caractère) : $11 \times 7 = 77$ bits

$\rightarrow$ 23 bits !



ABRACADABRA

Le codage de Huffman

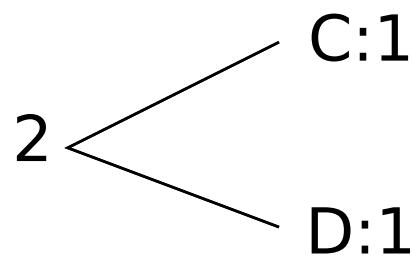
ABRACADABRA

A :5 B :2 R :2 C :1 D :1

Le codage de Huffman

ABRACADABRA

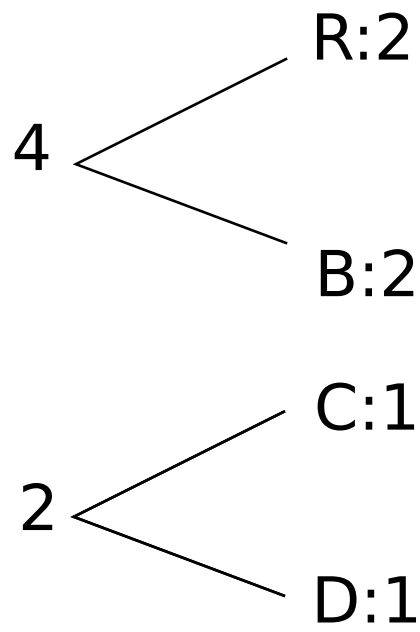
A :5 B :2 R :2



Le codage de Huffman

ABRACADABRA

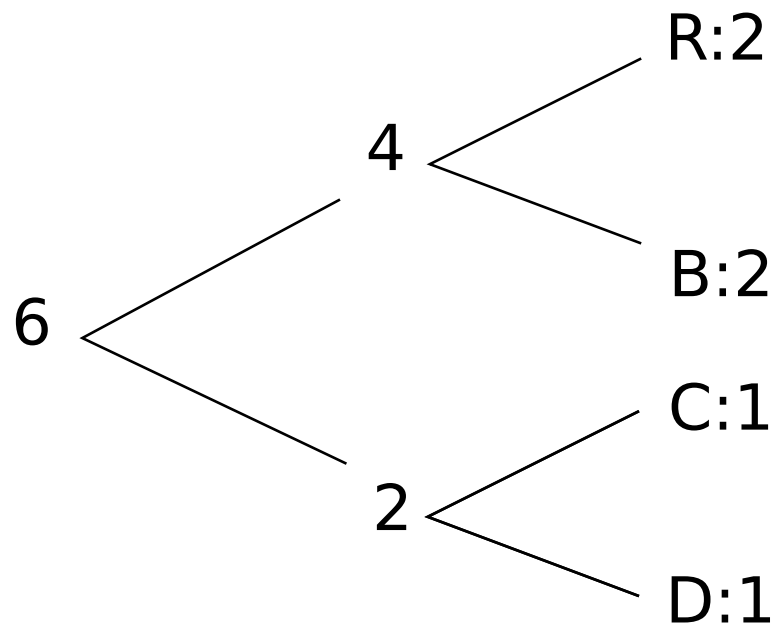
A :5



Le codage de Huffman

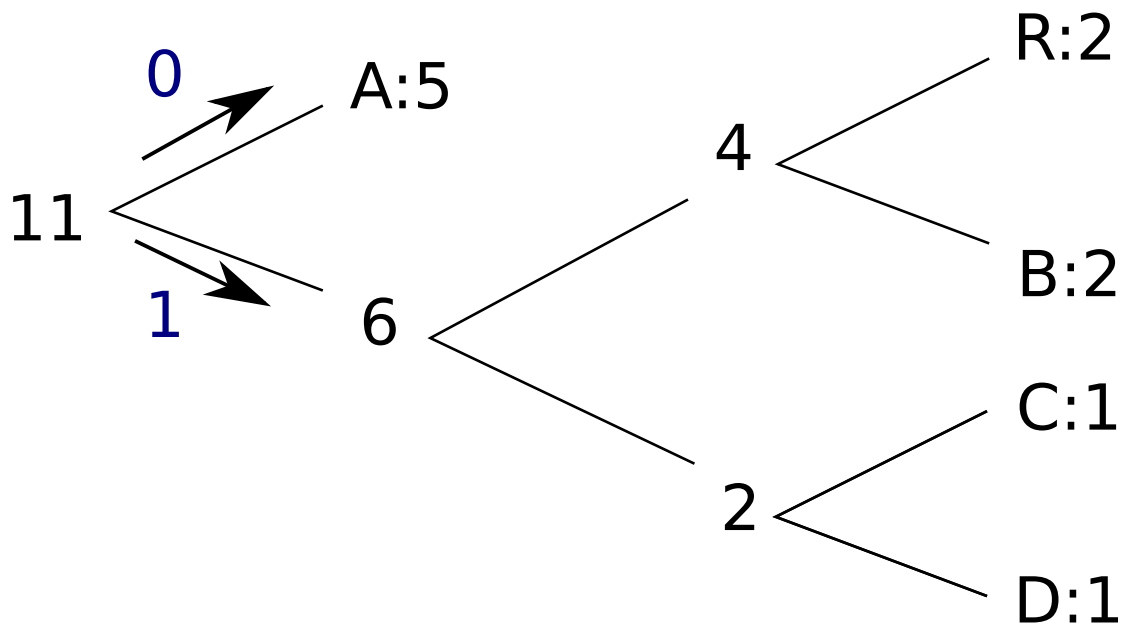
ABRACADABRA

A :5



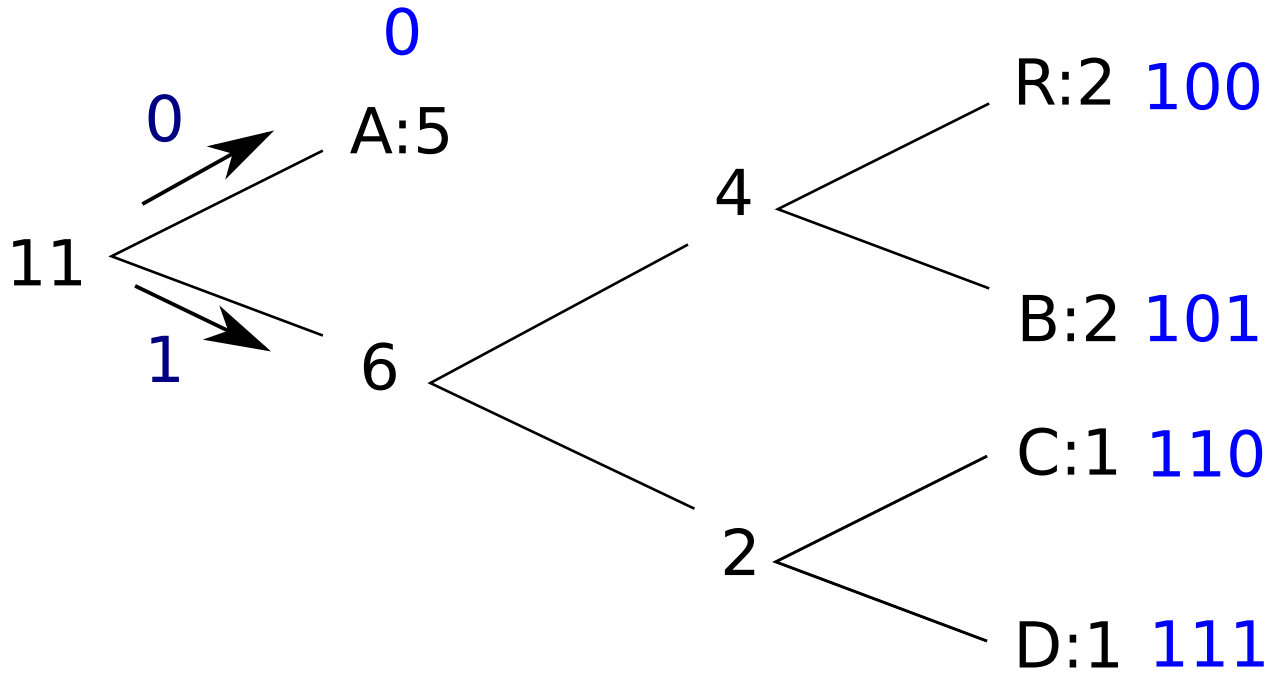
Le codage de Huffman

ABRACADABRA



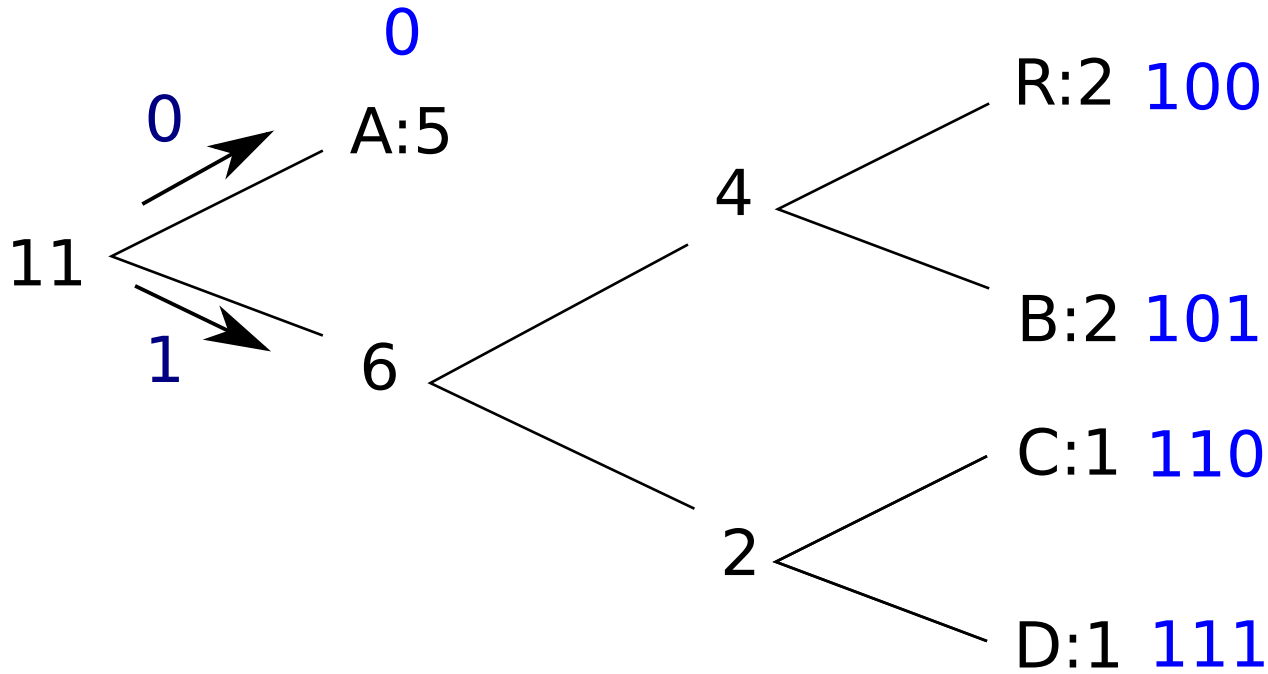
Le codage de Huffman

ABRACADABRA



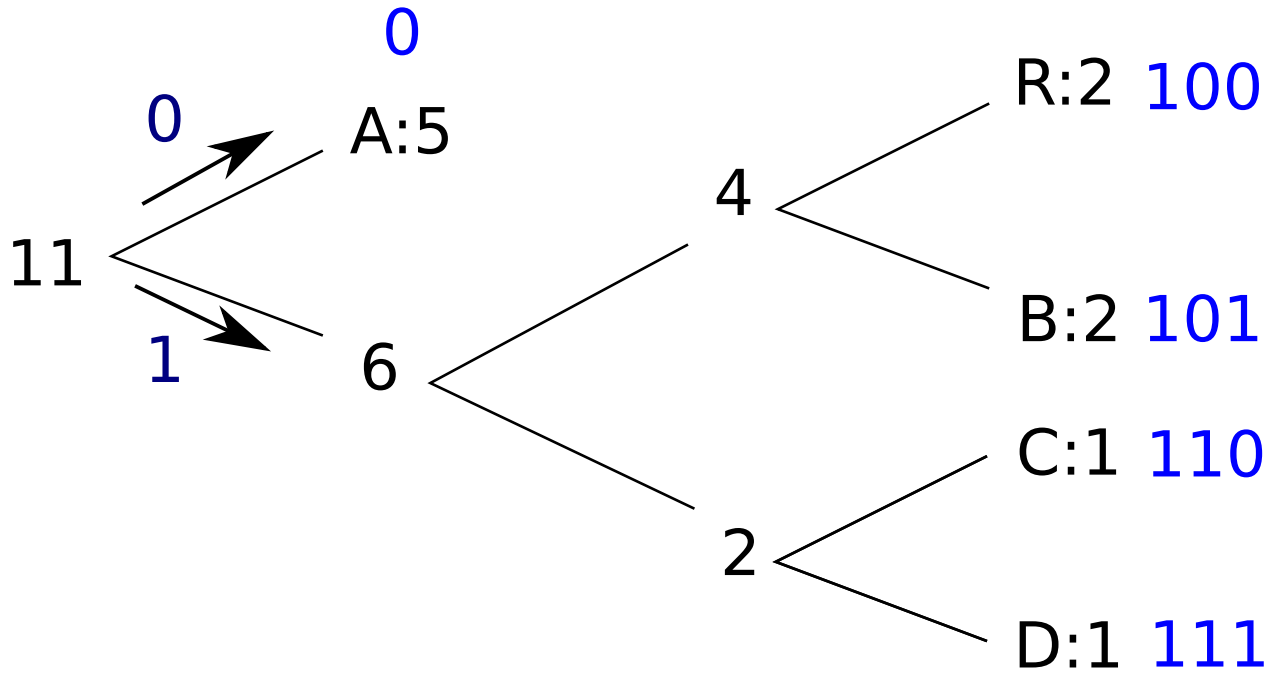
Le codage de Huffman

ABRACADABRA 0



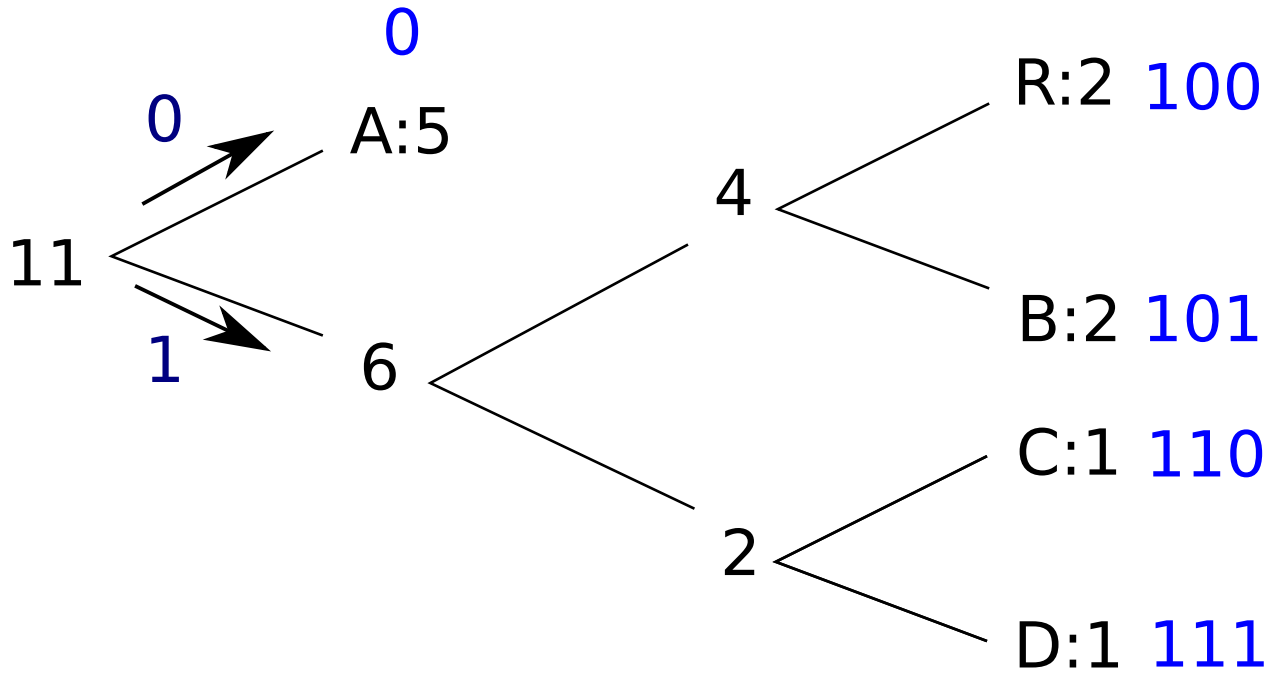
Le codage de Huffman

ABRACADABRA 0101



Le codage de Huffman

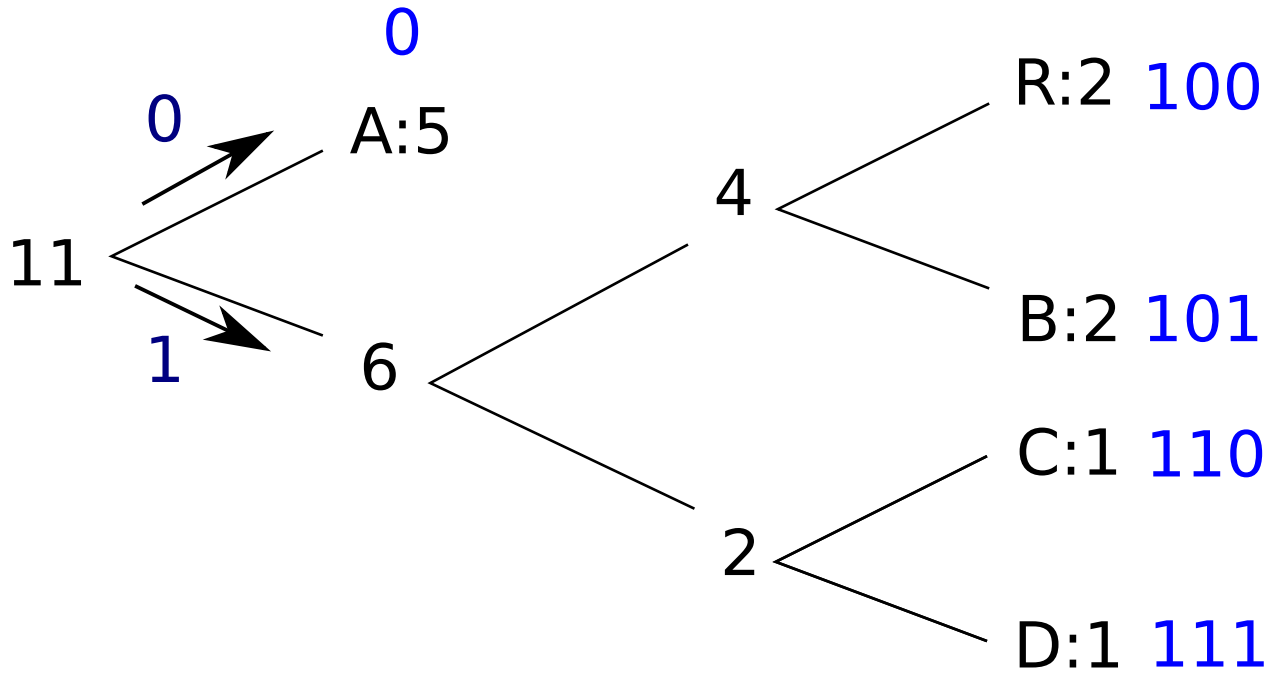
ABRACADABRA 01011000110011101011000



Le codage de Huffman

ABRACADABRA 01011000110011101011000

$\rightarrow$ 23 bits aussi !



Code par dictionnaire

Lempel-Ziv, Lempel-Ziv-Welch


- dictionnaire de base
- ajout de mots en fonction des motifs rencontrés

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB

A

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB

A

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR

AB

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA

ABRACAD

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA

ABRACAD

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA

ABRACAD

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA - ABR

ABRACAD(AB)

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA - ABR

ABRACAD(AB)

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA - ABR

ABRACAD(AB)(RA)

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA - ABR

ABRACAD(AB)(RA)

Le dictionnaire contient 136 entrées.

Code par dictionnaire

ABRACADABRA

- dictionnaire initialisé avec 128 caractères (codés sur 7 bits)
- ajout AB - BR - RA - AC - CA - AD - DA - ABR

ABRACAD(AB)(RA) 9 entrées

Le dictionnaire contient 136 entrées.