Codage et compression

Comment communiquer en dépensant moins ?

{Anne.Rasse,Jean-Marc.Vincent,Benjamin.Wack}@univ-grenoble-alpes.fr
{Maryline.Bruel,Herve.Barbe,Simon.Billouet}@ac-grenoble.fr

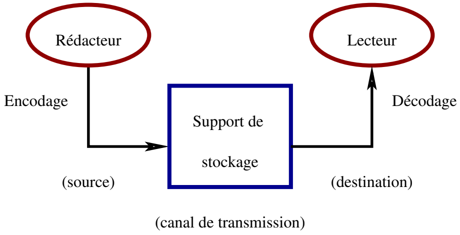
Maison Pour la Science : informatique débranchée

2017

CODAGE ET COMPRESSION

1. LE PROBLÈME : EFFICACITÉ D'UN CODE
2. COMPLEXITÉ D'UN CODE DE LONGUEUR VARIABLE
3. ALGORITHME DE HUFFMAN

Transmission de l'information



Critères de qualité d'un code :

- **Intégrité de l'information :** tolérance aux fautes (détection/correction des erreurs)
- **Sécurité de l'information :** authentification (cryptage)
- **Efficacité de la transmission :** compression des données

Donnée (message) : séquence finie de bits, éventuellement structurée

ACTIVITÉ DÉBRANCHÉE (1) : LES MARMOTTES

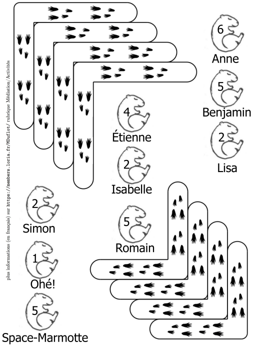
Un groupe de marmottes, moyennement satisfaites de leur terrier actuel, décide de concevoir unnouveau terrier et de le creuser avant l'hiver.

Règle1 : À partir de l'entrée on peut construire deux couloirs, et au bout de chaque couloir on peut faire un embranchement vers deux autres, mais pas plus (au risque de faire s'écrouler l'édifice).

Règle2 : Les marmottes vont chacune occuper une salle différente (pour ne pas se réveiller les unes les autres) et forcément une salle qui est tout au bout d'un couloir. Pour des marmottes au sommeil léger il est inenvisageable de dormir dans une salle à un embranchement, car les marmottes qui seraient au-delà de cet embranchement leur marcheraient dessus en entrant/sortant, et cela ruinerait leur hibernation

Règle 3 : Chaque marmotte se réveille un nombre précis de fois dans l'hiver, et s'il n'est pas grave de mettre assez loin de l'entrée une marmotte qui ne se réveille (et donc ne sort du terrier) qu'une fois dans l'hiver, c'est bien plus embêtant de mettre loin une marmotte qui va se réveiller 10 fois par exemple. Comme les pas de marmottes émettent de légères vibrations et que nos marmottes ont vraiment le sommeil léger, on va vouloir minimiser les déplacement de l'ensemble du groupe. Pour cela on va compter les déplacements de la façon suivante. Une marmotte dormant à 4 couloirs de l'entrée se réveillant 5 fois dans l'hiver va parcourir 4 x 5 = 20 couloirs aller et retour (pour simplifier on ne va compter que les aller-retour). On va faire la somme des déplacements de toutes les marmottes et essayer de rendre cette somme la plus petite possible.

Un grand merci à Marie Duflot
Plein d'activités



BARBARA⌣RASE⌣BLAISE⌣LE⌣BARBIER⌣!

ACTIVITÉ DÉBRANCHÉE (1) : LES MARMOTTES

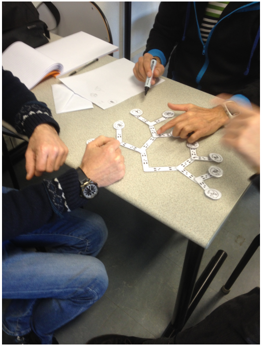
Un groupe de marmottes, moyennement satisfaites de leur terrier actuel, décide de concevoir unnouveau terrier et de le creuser avant l'hiver.

Règle1 : À partir de l'entrée on peut construire deux couloirs, et au bout de chaque couloir on peut faire un embranchement vers deux autres, mais pas plus (au risque de faire s'écrouler l'édifice).

Règle2 : Les marmottes vont chacune occuper une salle différente (pour ne pas se réveiller les unes les autres) et forcément une salle qui est tout au bout d'un couloir. Pour des marmottes au sommeil léger il est inenvisageable de dormir dans une salle à un embranchement, car les marmottes qui seraient au-delà de cet embranchement leur marcheraient dessus en entrant/sortant, et cela ruinerait leur hibernation

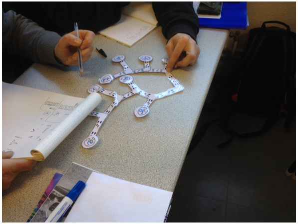
Règle 3 : Chaque marmotte se réveille un nombre précis de fois dans l'hiver, et s'il n'est pas grave de mettre assez loin de l'entrée une marmotte qui ne se réveille (et donc ne sort du terrier) qu'une fois dans l'hiver, c'est bien plus embêtant de mettre loin une marmotte qui va se réveiller 10 fois par exemple. Comme les pas de marmottes émettent de légères vibrations et que nos marmottes ont vraiment le sommeil léger, on va vouloir minimiser les déplacement de l'ensemble du groupe. Pour cela on va compter les déplacements de la façon suivante. Une marmotte dormant à 4 couloirs de l'entrée se réveillant 5 fois dans l'hiver va parcourir 4 x 5 = 20 couloirs aller et retour (pour simplifier on ne va compter que les aller-retour). On va faire la somme des déplacements de toutes les marmottes et essayer de rendre cette somme la plus petite possible.

À la pelle, on creuse



Activité débranchée (1a) : Les marmottes

Un premier terrier

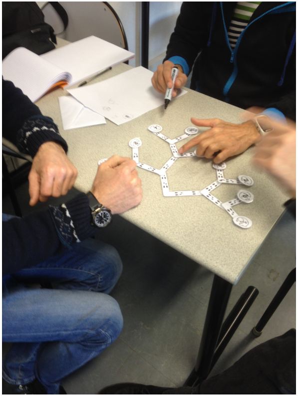


Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*G*	1
B	5	*DG*	2
R	5	*DDG*	3
⌣	5	*DDDG*	4
E	4	*DDDDG*	5
I	2	*DDDDDG*	6
L	2	*DDDDDDG*	7
S	2	*DDDDDDDG*	8
!	1	*DDDDDDDD*	8
total	32	3 – 5	

Bruit dans le terrier : 121 pour 32 réveils hivernaux
Déplacement moyen par réveil : $3,78$

Activité débranchée (1b) : Les marmottes

Un second terrier

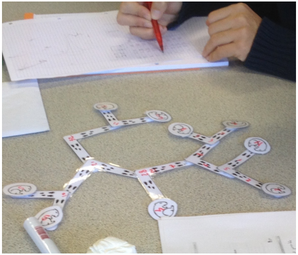


Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*G*	1
B	5	*DGG*	3
R	5	*DGD*	3
◡	5	*DDGG*	4
E	4	*DDGD*	4
I	2	*DDDG*	4
L	2	*DDDDG*	5
S	2	*DDDDDG*	6
!	1	*DDDDDD*	6
total	32	3 – 5	

Bruit dans le terrier : 108 pour 32 réveils hivernaux
Déplacement moyen par réveil : $3,375$

Activité débranchée (1c) : Les marmottes

Un troisième terrier

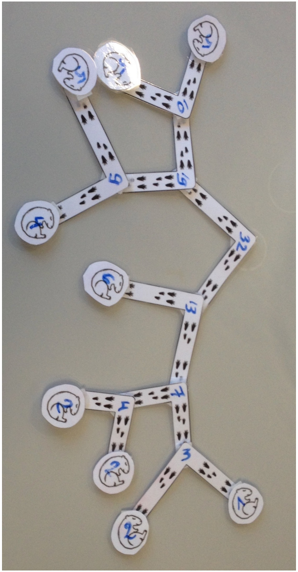


Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*GG*	2
B	5	*GD*	2
R	5	*DGG*	3
⌣	5	*DGD*	3
E	4	*DDG*	3
I	2	*DDDGG*	5
L	2	*DDDGD*	5
S	2	*DDDDG*	5
!	1	*DDDDD*	5
total	32	3 – 5	

Bruit dans le terrier : 99 pour 32 réveils hivernaux
Déplacement moyen par réveil : $3,09$

Activité débranchée (1d) : Les marmottes

Un quatrième terrier

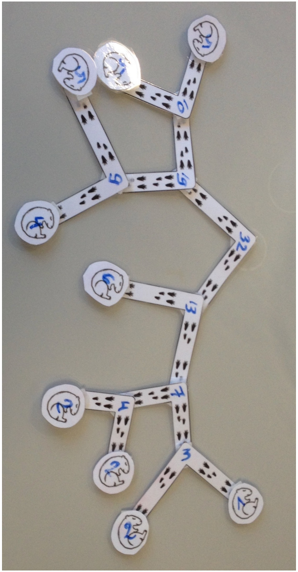


Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*DD*	2
B	5	*DGD*	3
R	5	*DGG*	3
$\smile$	5	*GDD*	3
E	4	*GDG*	3
I	2	*GGD*	3
L	2	*GGGD*	4
S	2	*GGGGD*	5
!	1	*GGGGG*	5
total	32	3 – 5	

Bruit dans le terrier : 98 pour 32 réveils hivernaux
Déplacement moyen par réveil : 3, 06

ACTIVITÉ DÉBRANCHÉE (1D) : LES MARMOTTES

Un quatrième terrier



Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*DD*	2
B	5	*DGD*	3
R	5	*DGG*	3
$\smile$	5	*GDD*	3
E	4	*GDG*	3
I	2	*GGD*	3
L	2	*GGGD*	4
S	2	*GGGGD*	5
!	1	*GGGGG*	5
total	32	3 – 5	

Bruit dans le terrier : 98 pour 32 réveils hivernaux
Déplacement moyen par réveil : 3, 06
Peut-on faire mieux ?
Y-a-t'il une limite ? (sans doute)
A-t'on une méthode pour construire une solution optimale ?

ACTIVITÉ DÉBRANCHÉE (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Activité débranchée (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Codage

Lettre	A	B	C	D	E		
Nombre	7	4	1	2	1	15	
Code1	000	001	010	011	100	code de longueur fixe	3.0

ACTIVITÉ DÉBRANCHÉE (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Codage

Lettre	A	B	C	D	E		
Nombre	7	4	1	2	1	15	
Code1	000	001	010	011	100	code de longueur fixe	3.0
Code2	0	1	10	11	100	code ambigü	1.3

Activité débranchée (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Codage

Lettre	A	B	C	D	E		
Nombre	7	4	1	2	1	15	
Code1	000	001	010	011	100	code de longueur fixe	3.0
Code2	0	1	10	11	100	code ambigü	1.3
Code3	0	10	1110	110	1111	code préfixe	1.9

Activité débranchée (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Codage

Lettre	A	B	C	D	E		
Nombre	7	4	1	2	1	15	
Code1	000	001	010	011	100	code de longueur fixe	3.0
Code2	0	1	10	11	100	code ambigü	1.3
Code3	0	10	1110	110	1111	code préfixe	1.9
Code4	01	11	110	00	010		2.1

Activité débranchée (2)

À l'aide de jetons transmettre le message suivant

ABABACADABEBAAD

Codage

Lettre	A	B	C	D	E		
Nombre	7	4	1	2	1	15	
Code1	000	001	010	011	100	code de longueur fixe	3.0
Code2	0	1	10	11	100	code ambigü	1.3
Code3	0	10	1110	110	1111	code préfixe	1.9
Code4	01	11	110	00	010		2.1

Trouver un code uniquement déchiffrable de longueur minimale.

CODAGE ET COMPRESSION

1. LE PROBLÈME : EFFICACITÉ D'UN CODE
2. COMPLEXITÉ D'UN CODE DE LONGUEUR VARIABLE
3. ALGORITHME DE HUFFMAN

COMPLEXITÉ D'UN CODE

Sources aléatoires : $p = (p_1, \cdots, p_k)$ fréquence de transmission ; longueur moyenne du codage

$$L(c) = \sum_{i=1}^{k} p_i l_i;$$

$$L_{inf} = \inf_c L(c); \ c \text{ ayant la propriété du préfixe}$$

COMPLEXITÉ D'UN CODE

Sources aléatoires : $p = (p_1, \cdots, p_k)$ fréquence de transmission ; longueur moyenne du codage

$$L(c) = \sum_{i=1}^{k} p_i l_i;$$

$$L_{inf} = \inf_c L(c); \ \ c \text{ ayant la propriété du préfixe}$$

Théorème (Shannon 1948)

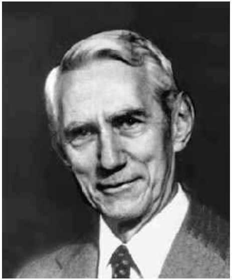
$$\mathcal{H}(p) \leqslant L_{inf} \leqslant \mathcal{H}(p) + 1;$$

avec

$$\mathcal{H}(p) = \sum_{i=1}^{k} p_i \log_2(\frac{1}{p_i});$$

quantité d'information contenue dans p (appelée également entropie).

CLAUDE SHANNON (1916-2001)



Claude Elwood Shannon (30 avril 1916 à Gaylord, Michigan - 24 février 2001), ingénieur électrique, est l'un des pères, si ce n'est le père fondateur, de la théorie de l'information. Son nom est attaché à un célèbre "schéma de Shannon" très utilisé en sciences humaines, qu'il a constamment désavoué.

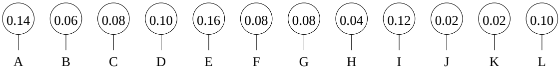
Il étudia le génie électrique et les mathématiques à l'Université du Michigan en 1932. Il utilisa notamment l'algèbre booléenne pour sa maîtrise soutenue en 1938 au MIT. Il y expliqua comment construire des machines à relais en utilisant l'algèbre de Boole pour décrire l'état des relais (1 : fermé, 0 : ouvert).
Shannon travailla 20 ans au MIT, de 1958 à 1978. Parallèlement à ses activités académiques, il travailla aussi aux laboratoires Bell de 1941 à 1972.
Claude Shannon était connu non seulement pour ses travaux dans les télécommunications, mais aussi pour l'étendue et l'originalité de ses hobbies, comme la jonglerie, la pratique du monocycle et l'invention de machines farfelues : une souris mécanique sachant trouver son chemin dans un labyrinthe, un robot jongleur, un joueur d'échecs (roi tour contre roi)...
Souffrant de la maladie d'Alzheimer dans les dernières années de sa vie, Claude Shannon est mort à 84 ans le 24 février 2001.

Wikipedia

Codage et compression

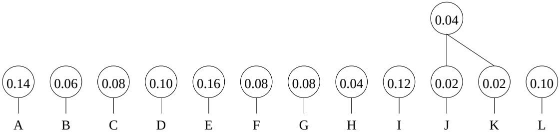
Algorithme de Huffman (1951)

A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

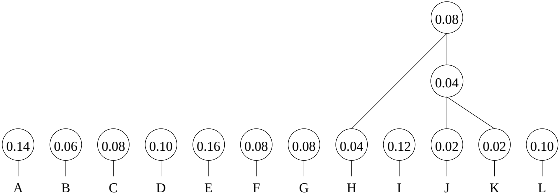


Algorithme de Huffman (1951)

A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

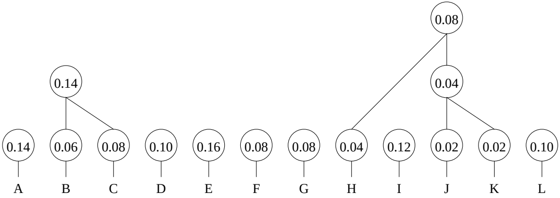


Algorithme de Huffman (1951)



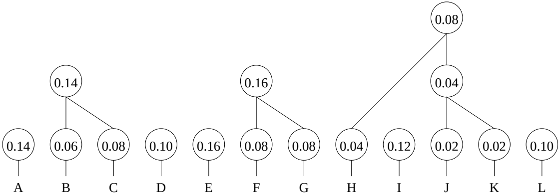
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)



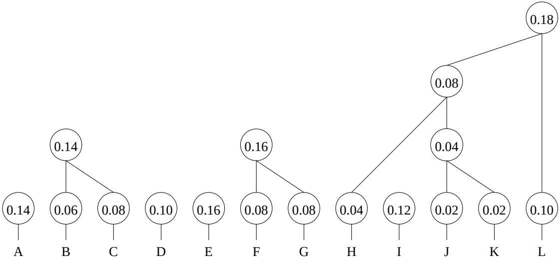
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)



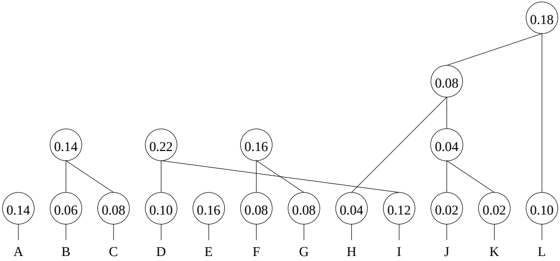
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

ALGORITHME DE HUFFMAN (1951)



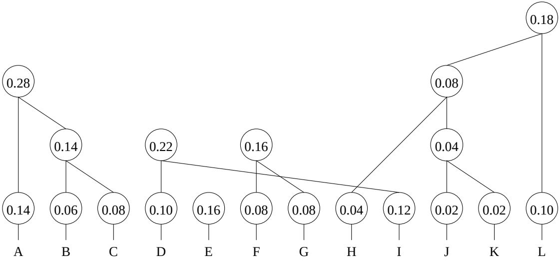
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

ALGORITHME DE HUFFMAN (1951)



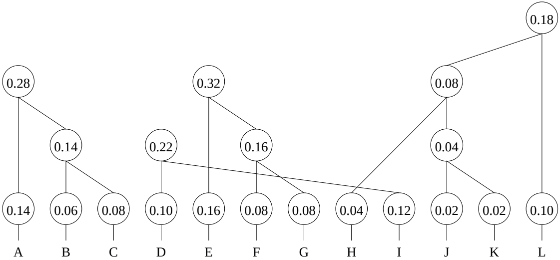
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)



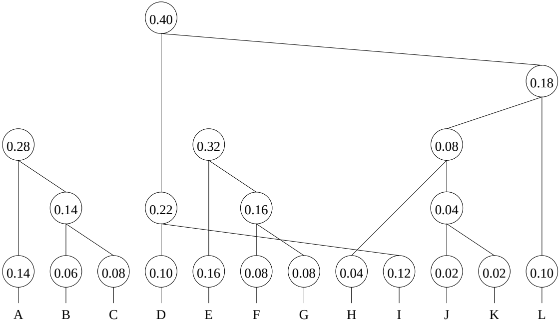
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

ALGORITHME DE HUFFMAN (1951)



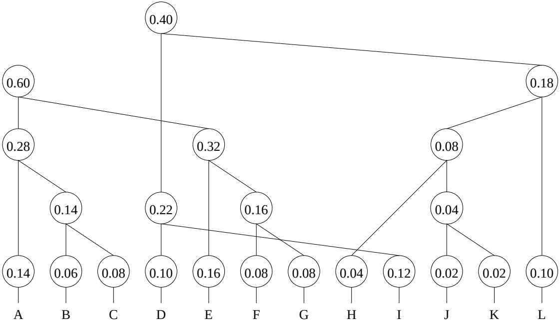
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

ALGORITHME DE HUFFMAN (1951)



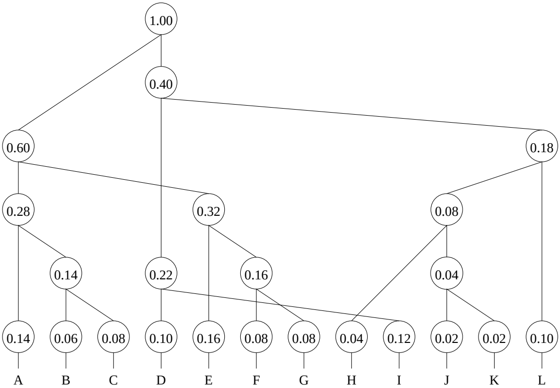
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)



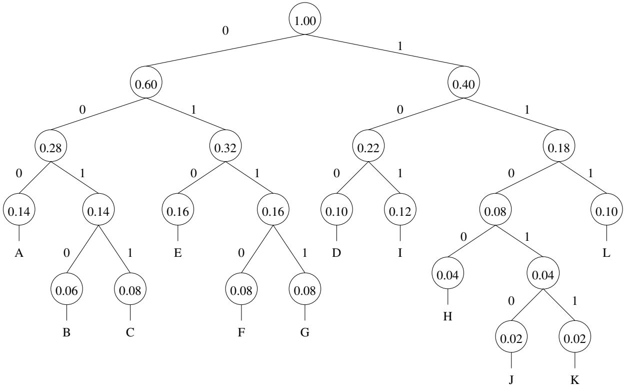
A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)



A	0.14
B	0.06
C	0.08
D	0.10
E	0.16
F	0.08
G	0.08
H	0.04
I	0.12
J	0.02
K	0.02
L	0.10

Algorithme de Huffman (1951)

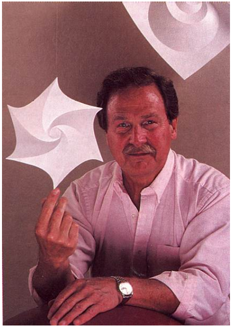


A	0.14	000
B	0.06	0010
C	0.08	0011
D	0.10	100
E	0.16	010
F	0.08	0110
G	0.08	0111
H	0.04	1100
I	0.12	101
J	0.02	11010
K	0.02	11011
L	0.10	111

Code optimal : L-inf = 3.42, Entropie = 3.38
Profondeur = $-\log_2$(probabilité)
Généralisation Lempel-Ziv(1978) et Lempel-Ziv-Welsh(1984)

DAVID A. HUFFMAN (1925-1999)

Avec ses sculptures



Huffman joined the faculty at MIT in 1953. In 1967, he went to University of California, Santa Cruz as the founding faculty member of the Computer Science Department. He played a major role in the development of the department's academic programs and the hiring of its faculty, and served as chair from 1970 to 1973. He retired in 1994, but remained active as an emeritus professor, teaching information theory and signal analysis courses.
Huffman made important contributions in many other areas, including information theory and coding, signal designs for radar and communications applications, and design procedures for asynchronous logical circuits. As an outgrowth of his work on the mathematical properties of "zero curvature Gaussian" surfaces, Huffman developed his own techniques for folding paper into unusual sculptured shapes (which gave rise to the field of computational origami).

http://www.huffmancoding.com/my-family/my-uncle/scientific-american

Huffman's algorithm (1951) : Implantation

```
HUFFMAN_ALGORITHM (p)
    Données: Ensemble de k symboles S et poids p
    Résultat: Optimal prefix code
    Node x,y,z
    File_à_Priorité F
    for s ∈ S
        z=new_node(p(s),/,/)
        Insère (F,z)
    for i = 1 to K − 1
        x=Extrait (F)
        y=Extrait (F)
        z=new_node(x.p+y.p,x,y)
        Insère (F,z)
    Return Extrait (F)
```

Algorithme de Huffman (1951) : Preuve

Invariant : La file a priorité contient une sous-forêt d'un arbre optimal de codage

Algorithme de Huffman (1951) : Preuve

Invariant : La file a priorité contient une sous-forêt d'un arbre optimal de codage

Initialisation

Cette précondition est vraie en début de l'algorithme.

ALGORITHME DE HUFFMAN (1951) : PREUVE

Invariant : La file a priorité contient une sous-forêt d'un arbre optimal de codage

Initialisation

Cette précondition est vraie en début de l'algorithme.

Preuve partielle

Si la précondition est vraie à l'entrée de l'itération, il existe un arbre optimal contenant la forêt incluse dans la file à priorité. Soit x et y les nœuds extraits de la FAP, il existe un arbre optimal tel que x et y soient 2 feuilles sœurs. Lorsque l'on réalise la fusion de x et y reste optimal.

ALGORITHME DE HUFFMAN (1951) : PREUVE

Invariant : La file a priorité contient une sous-forêt d'un arbre optimal de codage

Initialisation

Cette précondition est vraie en début de l'algorithme.

Preuve partielle

Si la précondition est vraie à l'entrée de l'itération, il existe un arbre optimal contenant la forêt incluse dans la file à priorité. Soit x et y les nœuds extraits de la FAP, il existe un arbre optimal tel que x et y soient 2 feuilles sœurs. Lorsque l'on réalise la fusion de x et y reste optimal.

Terminaison

L'algorithme, faisant un nombre fini d'itérations, se termine. On peut montrer que chaque itération diminue de 1 le nombre de nœuds dans la FAP. A la fin des itérations il ne reste qu'un nœud racine de l'arbre optimal.

Algorithme de Huffman (1951) : Preuve

Invariant : La file a priorité contient une sous-forêt d'un arbre optimal de codage

Initialisation

Cette précondition est vraie en début de l'algorithme.

Preuve partielle

Si la précondition est vraie à l'entrée de l'itération, il existe un arbre optimal contenant la forêt incluse dans la file à priorité. Soit x et y les nœuds extraits de la FAP, il existe un arbre optimal tel que x et y soient 2 feuilles sœurs. Lorsque l'on réalise la fusion de x et y reste optimal.

Terminaison

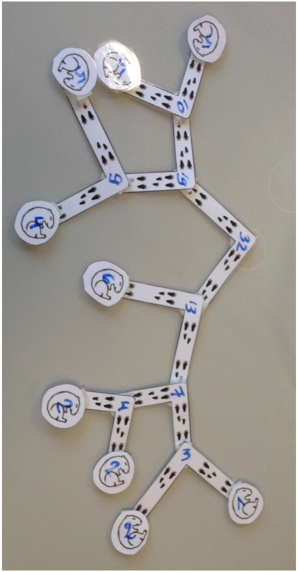
L'algorithme, faisant un nombre fini d'itérations, se termine. On peut montrer que chaque itération diminue de 1 le nombre de nœuds dans la FAP. A la fin des itérations il ne reste qu'un nœud racine de l'arbre optimal.

Complexité

Algorithme glouton : tout choix est définitif
Remarque : algorithme en $\mathcal{O}(k \log k)$

Activité débranchée (1e) : Les marmottes

Un terrier aux petits oignons

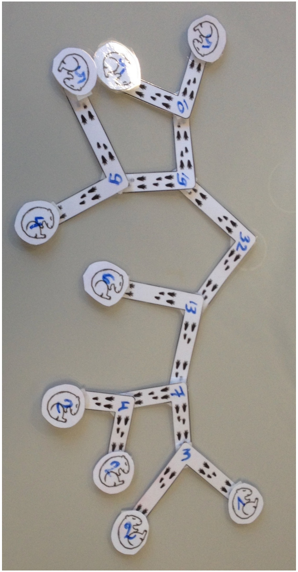


Lettre	*Nombre*	*Chemin*	*Long.*
A	6	*GD*	2
B	5	*DDD*	3
E	4	*DGG*	3
I	2	*GGDD*	4
L	2	*GGDG*	4
R	5	*DDG*	3
S	2	*GGGD*	4
$\smile$	5	*DGD*	3
!	1	*GGGG*	4
total	32	$2 - 5$	

Bruit dans le terrier : 97 pour 32 réveils hivernaux
Déplacement moyen par réveil : $3,03125$

Activité débranchée (1e) : Les marmottes

Un terrier aux petits oignons



Lettre	*Nombre*	*Chemin*	*Long.*	$log_2 \frac{1}{p_i}$
A	6	*GD*	2	2, 42
B	5	*DDD*	3	2, 68
E	4	*DGG*	3	3.00
I	2	*GGDD*	4	4.00
L	2	*GGDG*	4	4.00
R	5	*DDG*	3	2, 68
S	2	*GGGD*	4	4.00
␣	5	*DGD*	3	2, 68
!	1	*GGGG*	4	5.00
total	32	2 – 5		

Bruit dans le terrier : 97 pour 32 réveils hivernaux
Déplacement moyen par réveil : 3, 03125
Entropie associée 2, 989

Références

- **Introduction aux sciences de l'information**, Jean-Yves Le Boudec, Patrick Thiran et Rüdiger Urbanke, Presses polytechniques et universitaires romandes, 2015
- **Théorie des codes**, J-G. Dumas, J-L. Roch, E. Tannier et S. Varrette, Dunod 2007, Site
- **L'information : L'histoire - La théorie - Le déluge** James Gleick, Cassini 2015
- **Introductions aux Probabilités** Pierre Brémaud. Springer-Verlag, Berlin, 1984.
- **Algorithmique** Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein Dunod, 2011.
- **Incertitude et Information** Silviu Guiasu and Radu Theodorecu. Les Presses de l'Université LAVAL, Québec, 1971.
- **A mathematical theory of communication** Claude Elwood Shannon, Bells Systems Technical Journal, 27, 1948.

Sources pour les biographies MacTutor History