Faire la lumière sur les arbres digitaux

N. Broutin et L. Devroye

22 mars 2007

Algorithmique des mots

Qu'est-ce qu'on veut faire ?

- stocker les mots
- manipuler les mots

Les applications :

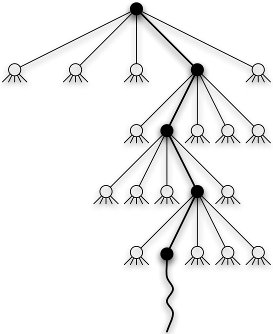
- compression
- bioinformatique
- traffic des réseaux

Alphabet $\{1, 2, \ldots, k\}$

- mots infinis
- probabilité de "i" est p_i

Les tries

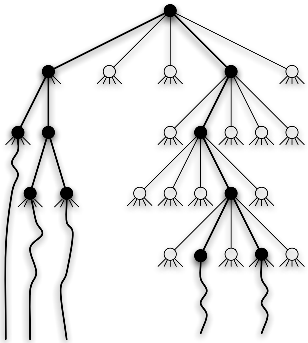
De la Briandais (1959)
Fredkin (1960)



- arbre de position
- mot / chemin
- n mots infinis indépendants
- dans les noeuds externes.

Les tries

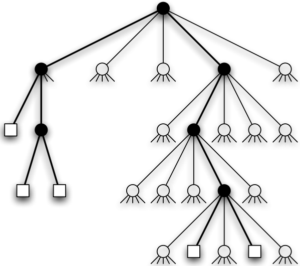
De la Briandais (1959)
Fredkin (1960)



- arbre de position
- mot / chemin
- n mots infinis indépendants
- dans les noeuds externes.

Les tries

De la Briandais (1959)
Fredkin (1960)



- arbre de position
- mot / chemin
- n mots infinis indépendants
- dans les noeuds externes.

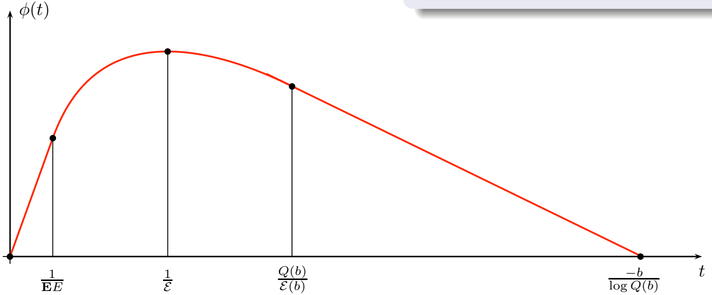
Tries aléatoires

Knuth (1973), Régnier, Pittel (1985)
Devroye, Szpankowski
Clément, Flajolet, Vallée
Hwang–Nicodème–Park–Szpankowski

$$\mathbf{E}P(t \log n) = n^{\phi(t)+o(1)}$$

Question:

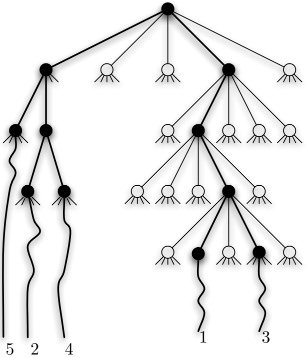
Explication des régimes ?



Arbres digitaux de recherche

Coffman–Eve (1970)
Konheim–Newman (1973), Pittel (1985)
Flajolet–Richmond (1992)
Louchard–Szpankowski–Tang (1995,1999)
Aguech–Lasmar–Mahmoud . . .

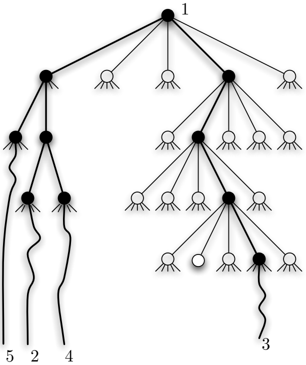
- réduire la taille (1)
- placer les mots dans les noeuds internes



Arbres digitaux de recherche

Coffman–Eve (1970)
Konheim–Newman (1973), Pittel (1985)
Flajolet–Richmond (1992)
Louchard–Szpankowski–Tang (1995,1999)
Aguech–Lasmar–Mahmoud ...

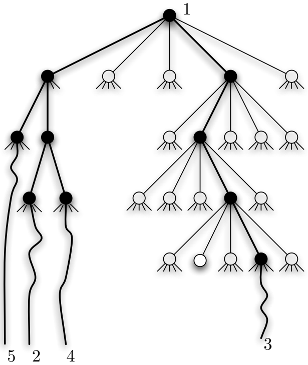
- réduire la taille (1)
- placer les mots dans les noeuds internes



Arbres digitaux de recherche

Coffman–Eve (1970)
Konheim–Newman (1973), Pittel (1985)
Flajolet–Richmond (1992)
Louchard–Szpankowski–Tang (1995,1999)
Aguech–Lasmar–Mahmoud …

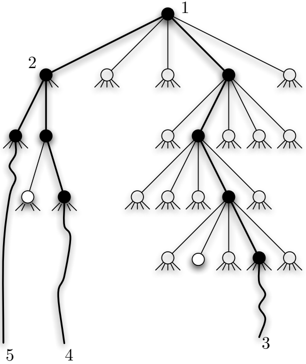
- réduire la taille (1)
- placer les mots dans les noeuds internes



Arbres digitaux de recherche

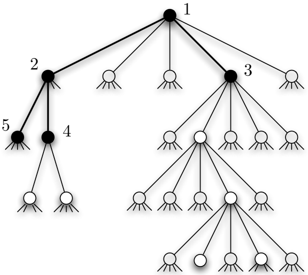
Coffman–Eve (1970)
Konheim–Newman (1973), Pittel (1985)
Flajolet–Richmond (1992)
Louchard–Szpankowski–Tang (1995,1999)
Aguech–Lasmar–Mahmoud ...

- réduire la taille (1)
- placer les mots dans les noeuds internes



Arbres digitaux de recherche

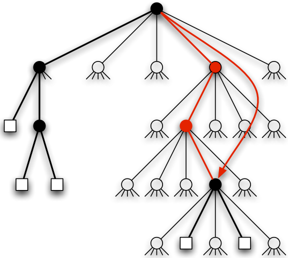
Coffman–Eve (1970)
Konheim–Newman (1973), Pittel (1985)
Flajolet–Richmond (1992)
Louchard–Szpankowski–Tang (1995,1999)
Aguech–Lasmar–Mahmoud . . .



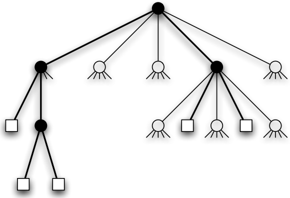
- réduire la taille (1)
- placer les mots dans les noeuds internes

Arbres PATRICIA

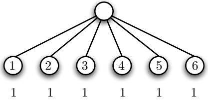
Morrison (1968)
Pittel, Bourdon
Devroye, Szpankowski

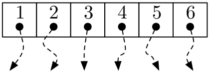


- réduire la taille (2)
- compacter les chemins



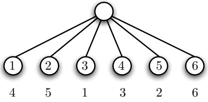
profondeur, hauteur / temps de recherche

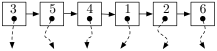




Question:

Et si l'implémentation ne repose pas sur un tableau ?





Objectifs

- expliquer les tries
- uniformiser la vision des arbres digitaux

Approche structurale

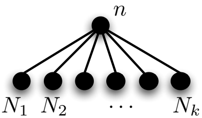
- temps de recherche maximum dans un trie hybride :

Tries pondérés

Vers une approche structurale

Distinction noeuds externes / noeuds internes :

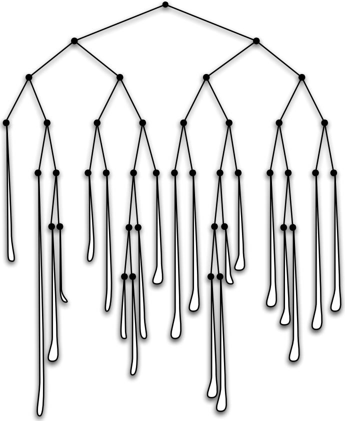
- utilitaire : les mots sont stockés dans les noeuds externes
- mais informe sur la structure ?



$$\left(\frac{N_1}{n}, \frac{N_2}{n}, \ldots, \frac{N_k}{n}\right) \xrightarrow[n\to\infty]{p.s.} (p_1, p_2, \ldots, p_k)$$

Le haut de l'arbre est déterministe !

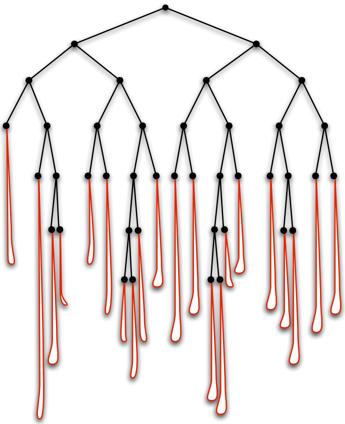
Le coeur et les spaghettis



Isoler la partie qu'on connaît :

- coeur : noeuds avec au moins $m(n) \to \infty$ mots, $m = o(\log n)$.
- spaghettis : ce qui pend.

Le coeur et les spaghettis



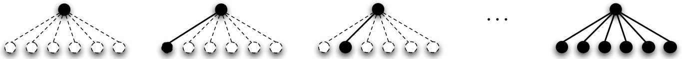
Isoler la partie qu'on connaît :

- coeur : noeuds avec au moins $m(n) \to \infty$ mots, $m = o(\log n)$.
- spaghettis : ce qui pend.

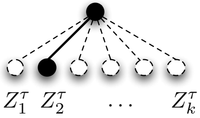
Le modèle : tries aléatoires pondérés

- forme de l'arbre : trie ordinaire
- 2^k différents types des noeuds $\tau \in \{0,1\}^k$:

$(0,0,0\dots,0)$ $(1,0,0\dots,0)$ $(0,1,0\dots,0)$ $\dots$ $(1,1,1\dots,1)$

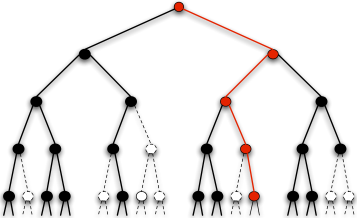


- Arêtes pondérées suivant le type : $\mathcal{Z}^\tau = (Z_1^\tau, Z_2^\tau, \dots, Z_k^\tau)$.



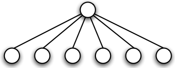
Modèle : tries aléatoires pondérés

- profondeur pondérée D_u d'un noeud u.
- marche aléatoire branchante sur un trie
- hauteur pondérée : $H_n = \max\{D_u : u \in T_n\}$.



Coeur vs Spaghetti

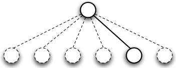
- dans le coeur, $N \to \infty$ donc :



poids $\simeq \mathcal{Z}^{(1,1,\ldots,1)}$

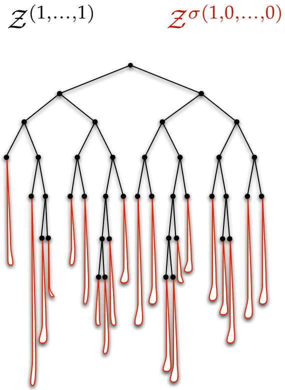
- dans les spaghettis,

  # mots $= o(\log n)$ donc : # noeuds de degré ≥ 2 est $o(\log n)$



poids $\simeq \mathcal{Z}^{\sigma(1,0,\ldots,0)}$

Hauteur pondérée : plan d'attaque



Comment compter les profondeurs ?

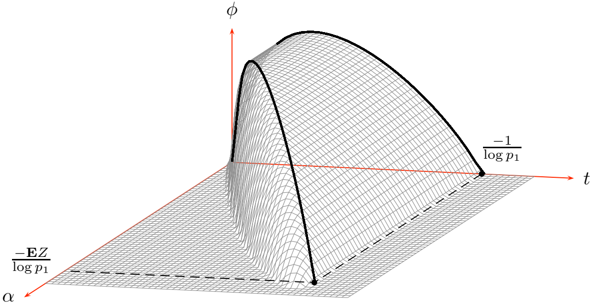
1. N : # noeuds au niveau k avec $D \simeq \ell$
2. S_N : hauteur du plus haut spaghetti parmi $N = N(k, \ell)$

$$H_n \simeq \sup_{k,\ell}\{\ell + S_N\}$$

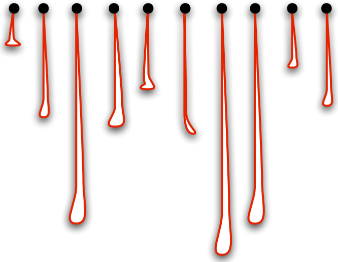
Le haut de l'arbre : le coeur

Profil pondéré $P(k, \ell)$:# noeuds au niveau k avec $D_u \simeq \ell$.

$$\phi(\alpha, t) = \lim_{n\to\infty} \frac{\log \mathbf{E}P(\alpha \log n, t \log n)}{\log n} \qquad \mathbf{E}P(\alpha \log n, t \log n) = n^{\phi(\alpha,t)+o(1)}$$



Le bas de l'arbre : les spaghettis



Forêt :

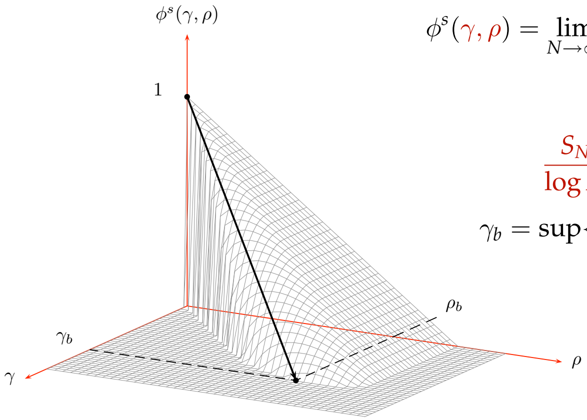
- Spaghettis nés au niveau $k \sim t \log n$
- avec $D_u \simeq \alpha \log n$

profil de la forêt :

$$P^s(\gamma \log n, \rho \log n) = \# \left\{ \begin{array}{l} \text{niveau } \rho \log n \\ D_v \simeq \gamma \log n \end{array} \right.$$

$N \simeq n^{\phi(\alpha,t)}$ tries indépendents tous de taille $\simeq m(n)$.

Les spaghettis

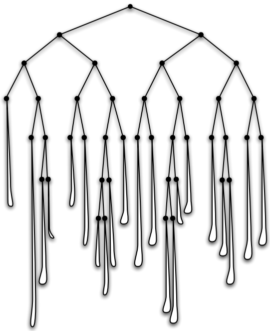


$$\phi^s(\gamma,\rho) = \lim_{N\to\infty} \frac{\log \mathbf{E} P^s(\gamma \log N, \rho \log N)}{\log N}$$

$$\frac{S_N}{\log N} \xrightarrow[n\to\infty]{} \gamma_b$$

$$\gamma_b = \sup\{\gamma : \phi^s(\gamma,\rho) > 0\}$$

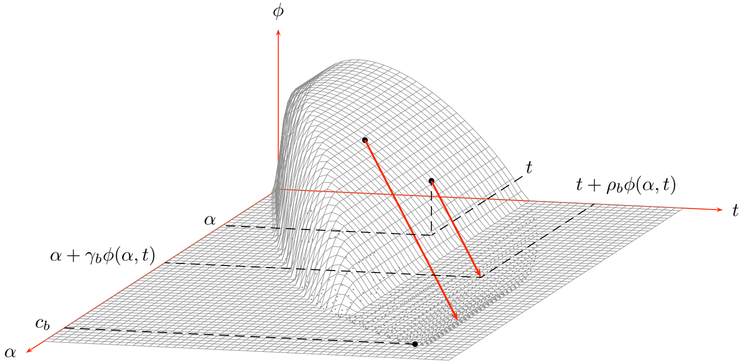
Hauteur pondérée



- pour $k = t \log n$ et $\ell = \alpha \log n$
- $N = n^{\phi(\alpha,t)}$
- $S_N \simeq \gamma_b \log N = \gamma_b \phi(\alpha, t) \log n$

$$\frac{H_n}{\log n} \xrightarrow[n\to\infty]{pr.} c = \sup\{\alpha + \gamma_b \phi(\alpha, t) : \phi(\alpha, t) > 0\}$$

Interprêtation géométrique



Arbres digitaux de recherche et PATRICIA

PATRICIA

- coeur : on coupe un noeud avec pr. $o(1)$
- spaghettis : on coupe tout sauf $o(\log n)$ noeuds par chemin

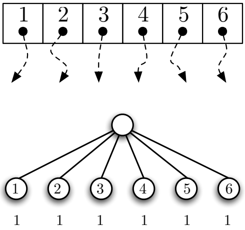
arbre digital de recherche

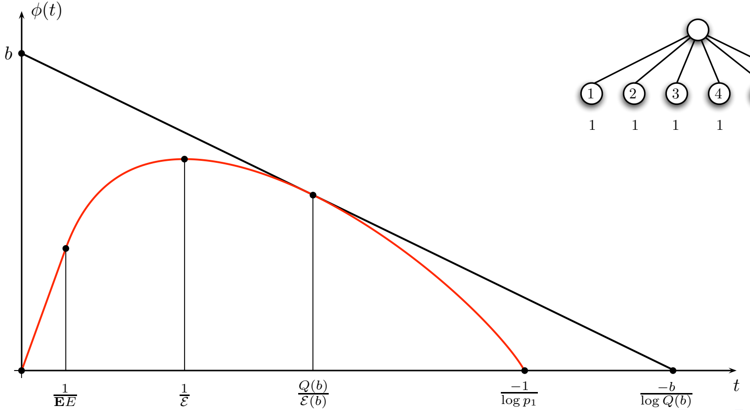
- spaghettis lineaires : pas de bouchon pour les élastiques
- coeur : croissance exponentielle les élastiques sont arêtés

Le profil est le même que celui du coeur

Exemple 1 : tries ordinaires

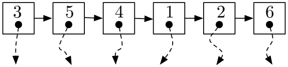
$$p_1 \geq p_2 \geq \cdots \geq p_k > 0$$

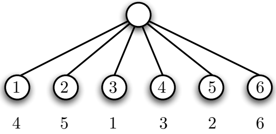




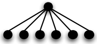
Exemple 2 : de la Briandais

de la Briandais (1959)
Clément, Flajolet et Vallée (1998, 2001)

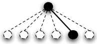




Comment pondérer les arêtes ?



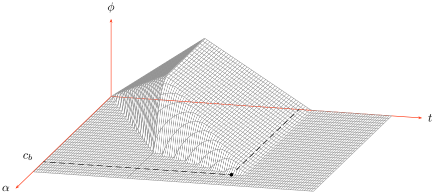
$\mathcal{Z} = (Z_1, Z_2, \ldots, Z_k)$



$\mathcal{Z}^s = (1, 1, \ldots, 1)$

de la Briandais, cas non-biaisé

$p_1 = p_2 = \cdots = p_k = 1/k$: permutation aléatoire

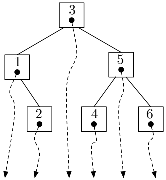


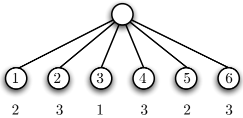
$$c = \frac{\log \sum_{i=1}^{k} k^{bi}}{b \log^2 k} \sim \frac{k}{\log k}$$

k	2	3	4	10	20
$c(k)$	3.72931...	3.03539...	3.03304...	4.36281...	6.68187...

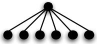
Exemple 3 : arbres ternaires de recherche

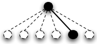
Clampett (1964)
Bentley et Sedgewick (1997)
Clément, Flajolet et Vallée (1998, 2001)





Distribution des poids :


$\mathcal{Z} = (Z_1, Z_2, \ldots, Z_k)$


$\mathcal{Z}^s = (1, 1, \ldots, 1)$

Arbres ternaires de recherche, cas non-biaisé

$p_1 = p_2 = \cdots = p_k = 1/k$: ABR aléatoire

Brown et Shubert (1984) : $$\mathbf{P}\{Z = \ell\} = \frac{2^{\ell-1}}{k \cdot k!} \sum_{j=\ell}^{k} \begin{bmatrix} k \\ j \end{bmatrix}$$

$$\frac{H_n}{\log n} \xrightarrow[n\to\infty]{} c \sim \frac{k \log(27/4)}{\log^2 k}$$

k	2	3	4	10	20
$c(k)$	3.72931...	2.89698...	2.72474...	3.05001...	3.88868...

Conclusions, et puis ?

- premier moment plus précis ?
- seconds moments ?
- sources Markoviennes, sources dynamiques ?
- arbres des suffixes ?