Arbres binaires de recherche

Brigitte Chauvin , Octobre 2001

Université de Versailles-Saint-Quentin
45 Avenue des Etats Unis, F-78035 Versailles Cedex
chauvin@math.uvsq.fr

Une référence sur le sujet est le livre de Mahmoud [7]

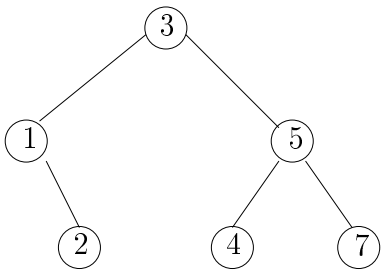
1 Définitions

Un **arbre binaire de recherche** est un arbre binaire dont chaque nœud interne est muni d'une "clé" (d'une étiquette, d'une marque, qui est dans un premier temps un entier) de telle sorte qu'à chaque nœud de l'arbre, toutes les clés du sous-arbre droit sont plus grandes que toutes les clés du sous-arbre gauche.

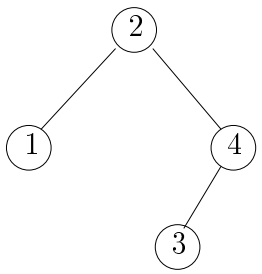
La définition (récursive) d'un **arbre binaire de recherche** est aussi la suivante : donnons-nous $x_1, x_2, \ldots x_n$ réels distincts. L'arbre binaire de recherche (abrégé abr) est l'arbre binaire dans lequel

- x_1 est à la racine
- le sous-arbre gauche est l'abr associé à $\{x_2, \ldots x_n\} \cap]-\infty, x_1[$
- le sous-arbre droit est l'abr associé à $\{x_2, \ldots x_n\} \cap]x_1, +\infty[$

Exemple : trier 3, 1, 5, 4, 7, 2



Remarque : la lecture de l'arbre en ordre symétrique donne la liste triée.
Attention : différentes permutations peuvent donner le même arbre. Par



exemple, 2 1 4 3 et 2 4 1 3 donnent l'arbre

Insertion : si l'on veut insérer une (n+1)ème donnée x_{n+1}, on la compare à la racine, puis à la clé racine du sous-arbre gauche (ou droit), etc... on ne la compare pas à toutes les clés mais seulement le long d'une branche, jusqu'à l'insérer sur un noeud externe. Appelons D_{n+1} (qui est plus ou moins un coût) le niveau d'insertion de la (n+1)ème donnée. Cet entier est toujours inférieur ou égal à $1 + h_n$, où h_n est la hauteur de l'arbre T_n :

$$h_n := \max_{u \in T_n} \mid u \mid .$$

Ces deux variables aléatoires sont intéressantes à étudier. On a toujours, en considérant les deux cas extrêmes

$$\lfloor log_2 n \rfloor \leq h_n \leq n - 1$$

et on verra plus loin que l'ordre de grandeur de la hauteur est heureusement en $\log n$.

Modèle probabiliste : à tout ensemble de n données distinctes $x_1, x_2, \ldots x_n$, on associe un arbre binaire de recherche T_n, à n noeuds internes et n+1 noeuds externes. Le modèle que nous étudions dans la suite, c'est-à-dire la loi de probabilité sur les arbres est l'image (par cette association) de la **loi uniforme sur les permutations** de n objets. Dans ce modèle, les arbres (T_n) ont même loi que ceux construits en choisissant $x_1, x_2, \ldots x_n$ selon une densité uniforme sur l'intervalle $[0, 1]$. Voyons, grâce au lemme suivant que dans ce modèle, l'insertion de la (n+1)ème donnée se fait uniformément sur les n+1 noeuds externes. Le lemme montre même que c'est le cas pour $x_1, \ldots, x_N$ i.i.d., de fonction de répartition $F(x)$.

Lemme 1 *Appelons U_k le rang de la donnée x_k parmi $x_1, \ldots, x_k$. Alors*

$$\mathbb{P}(U_k = j) = \frac{1}{k}$$

c'est-à-dire que U_k est de loi uniforme sur $\{1, \ldots, k\}$, quelle que soit la loi des données.

Démonstration Comme il s'agit d'une loi continue, la probabilité que x_k soit égale à l'une des données précédentes est nulle. De plus, dire que la k-ième donnée est à la place j, revient à dire que $j-1$ données sont plus petites et que $(k-1)-(j-1)$ données sont plus grandes, on a donc :

$$\begin{aligned}
\mathbb{P}(U_k = j) &= \int_{-\infty}^{+\infty} \mathbb{P}(U_k = j / x_k = x)\, dF(x) \\
&= \int_{-\infty}^{+\infty} \binom{k-1}{j-1} \big(F(x)\big)^{j-1} \big(1-F(x)\big)^{k-j} dF(x) \\
&= \int_0^1 \binom{k-1}{j-1} t^{j-1}(1-t)^{k-j}\, dt \\
&= \binom{k-1}{j-1} \beta(j, k-j+1) \\
&= \frac{1}{k}
\end{aligned}$$

□

Supposons construit un arbre à $n-1$ nœuds (internes) correspondants aux données $x_1, \ldots, x_{n-1}$. Numérotons les n nœuds externes disponibles de 1 à n de gauche à droite. Par lecture en ordre symétrique, on se rend compte que l'événement "$U_n = j$" est aussi l'événement "x_n est insérée sur le nœud externe numéro j". Une conséquence intéressante du lemme précédent est ainsi : après construction d'un arbre à $n-1$ noeuds, l'insertion d'une n-ième clé a la même probabilité d'être effectuée à chacun des n nœuds externes.

Attention : dans ce modèle probabiliste, appelé "modèle des permutations uniformes" (random permutation model), la loi uniforme est sur $\mathcal{S}_n$, le groupe des permutations ; ce qui signifie que les abr ne sont pas équiprobables, à l'inverse des arbres de Catalan.

2 Hauteur d'un abr ; connection abr-marche aléatoire de branchement

Les résultats sur la hauteur sont essentiellement dus à Devroye [3] et ils peuvent pour certains être compris, retrouvés ou améliorés à partir des résultats de Biggins [1] sur les marches aléatoires avec branchement.

Rappelons que T_n est l'abr de taille n, associé à $x_1, x_2, \dots x_n$. Pour tout nœud u de T_n, appelons $V_u =\mid T_u \mid$ la taille (c'est-à-dire le nombre de nœuds) du sous-arbre issu de u. Pour la racine (notée $\emptyset$), $V_\emptyset = n$ et si $V_1 =\mid T_1 \mid$, alors $V_2 =\mid T_2 \mid= n-1-V_1$. On sait que V_1 est uniforme sur $\{0, 1, \dots, n-1\}$.

Maintenant attachons à tous les nœuds u des variables aléatoires indépendantes U_u, de loi uniforme sur $[0, 1]$. Alors un calcul immédiat conduit au

Lemme 2 *: V_1 a même loi que $\lfloor nU \rfloor$; en corollaire, $n-1-V_1 = V_2$ a même loi que $\lfloor n(1-U) \rfloor$.*

De la même façon, on obtient que pour n'importe quel nœud u de l'arbre, de taille $V_u =\mid T_u \mid$, le sous-arbre gauche est de taille V_{u1} qui a même loi que $\lfloor V_u U_u \rfloor$ et le sous-arbre droit est de taille V_{u2} qui a même loi que $\lfloor V_u(1-U_u) \rfloor$. Finalement, pour un nœud à distance k de la racine, c'est-à-dire à profondeur k, V_u a même loi qu'un produit du type $\lfloor \dots \lfloor \lfloor nU \rfloor U_2 \rfloor \dots U_k \rfloor$ (où l'on écrit abusivement U_i au lieu de U_i pour un i-ème ancêtre qui est une branche gauche et $1 - U_i$ pour un i-ème ancêtre qui est une branche droite). Or, la loi de V_u est intéressante car la loi de la hauteur h_n de l'arbre est reliée aux V_u par la dualité suivante, qui est la base de la connection recherchée :

Proposition 1

$$\{h_n \geq k\} \;=\; \{\max_{u,|u|=k} V_u \geq 1\}.$$

Attention : le calcul de la loi de h_n ne sera pas simple, car les V_u sont très dépendants les uns des autres.

Reprenons le raisonnement précédent en termes de proportions, c'est-à-dire en considérant les $\frac{V_u}{n}$, qui représentent la proportion de nœuds dans le sous-arbre issu de u. Lorsque n tend vers $+\infty$, on obtient (démonstration rigoureuse à faire) que la proportion de nœuds dans le sous-arbre issu de u est

$$\prod_{v<u} U_v \ ,$$

où l'inégalité $v < u$ s'entend sur l'arbre, elle signifie "v ancêtre de u".

Considérons par ailleurs une marche aléatoire de branchement de processus ponctuel Z donné par

$$Z = \delta_{-\log U} + \delta_{-\log(1-U)}$$

($\delta_{\{k\}}$ désigne la mesure de Dirac au point k) où U suit une loi uniforme sur $[0,1]$. Dans ce modèle, la "position" de u dans l'arbre de branchement est donnée par

$$X_u = -\log(\prod_{v<u} U_v)$$

(avec la même convention : U_v peut être U_v ou $1 - U_v$), soit encore

$$e^{-X_u} = \prod_{v<u} U_v \ .$$

Ces égalités sont à comprendre en loi, car au pied de la lettre, on doit trouver dans le produit ci-dessus des variables U_v et $1 - U_v$, suivant si on a affaire à une branche à gauche ou à droite. Comme U_v et U_w sont indépendantes pour v différent de w et comme U_v et $1 - U_v$ sont chacune de loi uniforme sur $[0,1]$, cela n'induit pas de complication ici.

On voit ainsi que les deux modèles coïncident, du moins pour n assez grand. En guise d'application de cette connection entre les deux modèles, la proposition 1 induit

$$\{h_n \geq k\} \ = \ \{\max_{u,|u|=k} (\prod_{v<u} U_v) \geq \frac{1}{n}\}$$

$$\{h_n \geq k\} \ = \ \{\max_{u \in z_k} \ e^{-X_u} \geq \frac{1}{n}\}$$

ce qui signifie que le comportement asymptotique de la hauteur est donné par le comportement asymptotique de la position de la particule la plus à gauche dans la marche aléatoire de branchement. Or ce problème a été résolu dans ce dernier cadre et l'on obtient ainsi le

Théorème 1 *(Devroye, 1986)*

$$\frac{h_n}{\log n} \ \longrightarrow \ \textit{Constante,} \quad \textit{presque sûrement quand } n \to +\infty$$

De façon duale, introduisons h'_n le premier niveau d'apparition d'un noeud externe (c'est aussi le coût minimal d'une recherche vaine) ; h'_n s'appelle niveau de saturation de l'abr. Alors

Théorème 2

$$\frac{h_n}{\log n} \longrightarrow c \quad p.s. \qquad \frac{h'_n}{\log n} \longrightarrow c' \quad p.s$$

où c et c' sont solutions de l'équation

$$x \log \frac{2e}{x} = 1.$$

La conjecture dite de Robson disant que **la variance de la hauteur d'un arbre binaire de recherche est d'ordre de grandeur 1**, a longtemps tenu mais est maintenant résolue, et par plusieurs méthodes (Devroye, Reed [4], Drmota [5]).

3 Répartition des nœuds externes, largeur d'un abr.

3.1 Notations

Outre la hauteur et le niveau de saturation, si l'on veut connaître plus finement la répartition des noeuds par niveau dans l'arbre binaire de recherche, on introduit naturellement

$U_k(n) :=$ le nombre de noeuds externes au niveau k dans l'arbre T_n.

On pourrait travailler de façon analogue avec les noeuds internes et introduire

$V_k(n) :=$ le nombre de noeuds internes au niveau k dans l'arbre T_n.

ainsi que

$$\begin{aligned} Z_k(n) &:= \text{ le nombre total de noeuds au niveau } k \text{ dans l'arbre } T_n \\ Z_k(n) &= U_k(n) + V_k(n). \end{aligned}$$

Comme il y a $n+1$ noeuds externes dans l'arbre T_n de taille n, la mesure empirique de répartition des noeuds externes est

$$\nu_n := \sum_{k=1}^{+\infty} \frac{U_k(n)}{n+1}\ \delta_{\{k\}}.$$

Comme la $(n+1)$-ième insertion se fait uniformément sur les $n+1$ noeuds externes de l'arbre T_n, la loi du niveau d'insertion, ou profondeur D_n est donnée par

$$\mathbb{P}(D_{n+1} = k \mid T_n) = \frac{U_k(n)}{n+1} = \nu_n(k), \tag{1}$$

et en prenant l'espérance

$$\mathbb{P}(D_{n+1} = k) = \mathbb{E}(\frac{U_k(n)}{n+1}) = \mathbb{E}(\nu_n(k))$$

autrement dit la loi de D_{n+1} est ce qu'on appelle la mesure d'intensité (déterministe) du processus ponctuel ν_n, par conséquent **les résultats en moyenne sur la mesure ν_n donneront des résultats en loi sur D_n.**

3.2 Résultats en moyenne sur les nœuds externes

Théorème 3 *(Lynch, 1965)*

$$\mathbb{E}U_k(n) = \frac{2^k}{n!}\begin{bmatrix} n \\ k \end{bmatrix}$$

où $\begin{bmatrix} n \\ k \end{bmatrix}$ sont les nombres de Stirling de première espèce, c'est-à-dire que $\begin{bmatrix} n \\ k \end{bmatrix}$ est le coefficient de x^k dans le développement de $x(x+1)\dots(x+n-1)$. Ces nombres vérifient la relation de récurrence

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + n\begin{bmatrix} n \\ k \end{bmatrix}.$$

Corollaire 1 *La loi de D_n est donnée par*

$$\mathbb{P}(D_{n+1} = k) = \frac{2^k}{(n+1)!}\begin{bmatrix} n \\ k \end{bmatrix}$$

Preuve du théorème 3

On utilise fortement une sorte de propriété de branchement, ici le conditionnement par la taille du sous-arbre gauche T_1. Posant

$$a_{n,k} := \mathbb{E}U_k(n)\ \frac{n!}{2^k},$$

un petit calcul permet de voir que les $a_{n,k}$ satisfont la relation de récurrence des nombres de Stirling et ont les mêmes valeurs initiales. □

3.3 Résultats p.s. sur les nœuds externes

On voit facilement que pour tout niveau k de l'arbre T_n ,

$$Z_{k+1}(n) = 2V_k(n)$$

et donc

$$Z_{k+1}(n) - Z_k(n) = V_k(n) - U_k(n)$$

Comme au plus bas niveau de l'arbre, il n'y a que des nœuds externes

$$U_{1+h_n}(n) = Z_{1+h_n}(n) \text{ et } V_{1+h_n}(n) = 0$$

et par une récurrence descendante sur k :

$$Z_k(n) = \sum_{j=k}^{1+h_n} \frac{U_j(n)}{2^{j-k}}$$

qui est une quantité du type "polynôme de niveau".

Définition 1 *On appelle* **polynôme de niveau** *d'un arbre binaire de recherche T_n, le polynôme défini pour tout paramètre $z \in \mathbb{C}$ par*

$$W_n(z) := \sum_{k=0}^{+\infty} U_k(n)\ z^k$$

Il s'agit bien d'un polynôme, car pour $k > 1 + h_n, U_k(n) = 0$. Bien entendu, c'est une variable aléatoire, puisque les $U_k(n)$ sont aléatoires.

3.4 Résultats en moyenne sur le polynôme de niveau

On obtient très facilement la moyenne du polynôme de niveau, car

$$\mathbb{E}W_n(z) = \sum_{k=0}^{+\infty} \mathbb{E}U_k(n)\ z^k$$

et d'après le théorème 3,

$$\begin{aligned} \mathbb{E}W_n(z) &= \frac{1}{n!} \sum_{k=0}^{+\infty} 2^k z^k \begin{bmatrix} n \\ k \end{bmatrix} \\ &= \frac{1}{n!} 2z(2z+1)\dots(2z+n-1) \\ &= \prod_{j=0}^{n-1} \frac{j+2z}{j+1}\ . \end{aligned}$$

La loi de la profondeur est alors immédiate, car si $D_n(z)$ est sa série génératrice

$$D_n(z) = \sum_{k\geq 0} \mathbb{P}(D_n = k) z^k,$$

comme $\mathbb{P}(D_{n+1} = k) = \frac{1}{n+1}\mathbb{E}U_k(n)$ (cf. equation 1), on obtient

$$D_{n+1}(z) = \frac{1}{n+1}\mathbb{E}W_n(z) = \frac{1}{(n+1)!} \prod_{j=0}^{n-1} (j+2z)$$

qui est une expression assez explicite de la série génératrice de la profondeur d'insertion.

Exercice En déduire que

$$\mathbb{E}(D_{n+1}) = 2(H_{n+1} - 1)$$

et calculer $Var(D_{n+1})$.

3.5 Résultats p.s. sur le polynôme de niveau

La relation évidente

$$U_k(n+1) - U_k(n) = -\mathbb{1}_{\{D_{n+1}=k\}} + 2\ \mathbb{1}_{\{D_{n+1}=k-1\}}$$

permet d'avoir une relation de récurrence sur le "polynôme de niveau" :

$$\begin{aligned}\mathbb{E}(W_{n+1}(z) \mid T_n) &= \sum_{k\geq 0} z^k \Big[U_k(n) + 2\mathbb{P}(D_n = k-1 \mid T_n) - \mathbb{P}(D_n = k \mid T_n)\Big] \\ &= \frac{n+2z}{n+1}\, W_n(z)\ ,\end{aligned}$$

ce qui signifie que le polynôme de niveau convenablement renormalisé est une martingale.

Théorème 4 *pour tout nombre complexe $z \in \mathbb{C}$,*

$$M_n(z) := \frac{W_n(z)}{\mathbb{E}(W_n(z))} = \frac{W_n(z)}{\prod_{j=0}^{n-1} \frac{j+2z}{j+1}}$$

est une $\mathcal{F}_n$-martingale. Elle converge p.s. pour tout z réel positif. Elle converge dans L^1 sur l'intervalle réel $]c'/2; c/2[$ (les constantes c et c' étant celles du théorème 1). Elle converge dans L^2 sur la boule $B(1, \frac{1}{\sqrt{2}})$ de $\mathbb{C}$.

Idées de démonstration

La propriété de martingale vient du calcul ci-dessus. La convergence presque sûre dans $\mathbb{R}_+$ est celle de toute martingale positive. La convergence dans L^1 est plus difficile à voir. Elle est obtenue en bornant $M_n(z)$ dans L^p pour $p > 1$, de façon assez analogue à la méthode utilisée pour un processus de branchement spatial. Les détails sont dans [6]. La convergence L^2 pour z dans $\mathbb{C}$ s'obtient en calculant la covariance de la martingale et avec un peu d'analyse complexe. La convergence L^1 (et L^p) est un problème ouvert dans $\mathbb{C}$. Les détails se trouvent dans [2]. □

Globalement, cette étude tire parti de l'égalité

$$W_n(z) = \mathbb{E}(W_n(z))M_n(z)$$

qui permet d'étudier le polynôme de niveau en séparant la partie déterministe (son espérance qui est bien connue, cf. section précédente) et la partie aléatoire qui, étant une martingale, possède des propriétés agréables. C'est ainsi que des résultats de type théorème central limite et grandes déviations sur la mesure ν_n peuvent être obtenus, ainsi que l'ordre de grandeur de la largeur d'un arbre binaire de recherche. Appelons en effet $\overline{Z}_n$ la largeur de

T_n ; c'est le maximum sur tous les niveaux du nombre de noeuds à chaque niveau :

$$\overline{Z}_n := \max_{k \geq 0} \; Z_k(n)$$

Théorème 5 *(largeur d'un arbre binaire de recherche)*

$$\frac{\overline{Z}_n}{n/\sqrt{\pi \log n}} = 1 + \mathcal{O}(\frac{1}{\sqrt{\log n}}) \quad p.s.$$

lorsque $n \to +\infty$.

4 Coût de construction d'un abr ; coût de recherche ; longueurs de cheminement.

4.1 Coût de construction ; longueur de cheminement interne

Question : combien de clés regarde-t-on avant de placer une nouvelle clé à la bonne place ? Dans un arbre dégénéré (correspondant au cas le pire), on examine $i-1$ clés avant d'insérer la i-ième clé donc on examine $N(N-1)/2$ nœuds pour construire un arbre de N noeuds. Dans le cas le meilleur, le nombre milieu est placé à la racine pour chaque sous-arbre, chaque sous-arbre est de taille $N/2$ et donc le coût de construction de l'arbre vérifie la relation (du type diviser pour régner) $T_N = 2T_{N/2} + N$ ce qui donne $T_N \sim N \log N$ comme nombre d'étapes pour construire l'arbre.

Etant donné un arbre binaire de recherche (construit ou partiellement construit), on aura soit à insérer une nouvelle clé, soit à chercher une clé, d'où un coût de construction et un coût de recherche, mesurés en nombre de comparaisons. Dans les deux cas, cela se fera par un unique chemin dans l'arbre, parcouru par un pointeur partant de la racine et qui chemine en éliminant à chaque comparaison avec une clé racine, le sous-arbre droit ou le sous-arbre gauche. On voit ainsi que le coût de construction d'un abr de taille n, compté en nombre de comparaisons, n'est autre que la longueur de cheminement interne I_n.

$$I_n = \sum_{u \in T_n} \mid u \mid \; .$$

Pour étudier la loi de I_n, on introduit assez naturellement la fonction génératrice bivariée

$$F(x,y) = \sum_{n,k\geq 0} \mathbb{P}(I_n = k) x^n y^k \ .$$

Comme d'habitude, on fait opérer une sorte de propriété de branchement pour obtenir une relation de récurrence sur I_n. En effet, appelons, en omettant l'indice n, T_1 et T_2 les sous-arbres respectivement gauche et droit de T_n. Si T_1 est de taille p, alors T_2 est de taille $n-1-p$, $p = 0, \ldots, (n-1)$, et

$$I_n = I_p(T_1) + I_{n-1-p}(T_2) + (n-1) \tag{2}$$

et donc par l'indépendance conditionnelle des sous-arbres gauche et droit

$$\begin{aligned}
\mathbb{P}(I_n = k) &= \sum_{p=0}^{n-1} \mathbb{P}(\mid T_1 \mid = p) \mathbb{P}(I_n = k \ / \mid T_1 \mid = p) \\
&= \sum_{p=0}^{n-1} \frac{1}{n} \ \mathbb{P}(I_p(T_1) + I_{n-1-p}(T_2) + (n-1) = k \ / \mid T_1 \mid = p) \\
&= \sum_{p=0}^{n-1} \frac{1}{n} \sum_{k_1+k_2 = k-(n-1)} \mathbb{P}(I_p = k_1) \mathbb{P}(I_{n-1-p} = k_2)
\end{aligned}$$

ce qui donne l'équation différentielle suivante

$$\frac{\partial F(x,y)}{\partial x} = F^2(xy, y) \ . \tag{3}$$

Cette équation informe en partie sur la loi de I_n, par exemple, en la reliant aux fonctions génératrices des moments de I_n. Faisons-le pour l'espérance (pour la variance, les calculs sont plus lourds). Appelons B_1 la fonction génératrice de la moyenne de I_n.

$$B_1(x) := \sum_{n\geq 0} \mathbb{E}(I_n) \ x^n$$

Alors

$$B_1(x) = \frac{\partial F(x,y)}{\partial y} \mid_{y=1}$$

et en dérivant par rapport à x

$$\begin{aligned} B_1'(x) &= \frac{\partial}{\partial y} F^2(xy, y) \mid_{y=1} \\ &= 2xF(x,1)F^2(x,1) + 2F(x,1)B_1(x) \end{aligned}$$

Or,

$$F(x,1) = \sum_{n,k \geq 0} \mathbb{P}(I_n = k) x^n = \frac{1}{1-x}$$

ce qui montre que B_1 est solution de l'équation différentielle

$$B_1'(x) = \frac{2x}{(1-x)^3} + \frac{2B_1(x)}{1-x} \tag{4}$$

équation qui se résoud sans problème, en rajoutant les conditions initiales $I_0 = 0$ et $I_1 = 0$:

$$B_1(x) = \frac{-2}{(1-x)^2} \Big[x + \log(1-x) \Big] \ .$$

Pour obtenir $\mathbb{E}(I_n)$, il suffit alors d'extraire le coefficient de x^n dans $B_1(x)$. On peut par exemple écrire $B_1(x)$ comme le produit de deux séries

$$\begin{aligned} B_1(x) &= \frac{2}{(1-x)^2} \Big[\frac{x^2}{2} + \frac{x^3}{3} + \cdots + \frac{x^n}{n} + \ldots \Big] \\ &= 2x^2 \Big[\sum_{j \geq 0} (j+1) x^j \Big] \Big[\sum_{k \geq 0} \frac{1}{k+2} x^k \Big] \end{aligned}$$

et le coefficient de x^n est donc

$$\mathbb{E}(I_n) = 2 \sum_{k=0}^{n} \frac{n-k+1}{k+2}$$

qui s'écrit encore

$$\mathbb{E}(I_n) = 2(n+1)H_n - 4n = 2(n+1)(H_{n+1} - 1) - 2n$$

où H_n désigne la somme des inverses des n premiers entiers.

Remarque

On peut trouver directement le coût de construction moyen d'un abr de taille n, c'est-à-dire la longueur de cheminement interne moyenne $\mathbb{E}(I_n)$, en voyant que celle-ci vérifie l'équation de récurrence déduite de l'équation (2)

$$\mathbb{E}(I_n) = n - 1 + \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}(I_{k-1}) + \mathbb{E}(I_{n-k})$$

que l'on reconnaît, c'est l'équation de récurrence de l'algorithme de tri rapide. Elle se résoud en

$$\mathbb{E}(I_n) = 2(n+1)(H_{n+1} - 1) - 2n$$

où H_n désigne la somme des inverses des n premiers entiers. Cela montre que le coût moyen de construction d'un abr est d'ordre de grandeur $n \log n$ (comme dans le cas le meilleur !).

4.2 Coût de recherche dans un abr ; longueur de cheminement externe

Si E_n est la longueur de cheminement externe, c'est-à-dire la somme des longueurs de u pour tous les u noeuds externes de l'arbre T_n et si I_n est la longueur de cheminement interne, alors ces deux variables sont reliées par

Lemme 3

$$E_n = I_n + 2n$$

Démonstration : par récurrence sur n, en détaillant le mécanisme de passage de T_n à T_{n+1}. □

On dit qu'une recherche est réussie quand la clé objet de la recherche est trouvée. On dit que la recherche est vaine si elle se termine sur un noeud externe. On appelle D_n la profondeur (le niveau) d'insertion de la n-ième donnée, c'est aussi le coût d'une recherche vaine. Il peut arriver selon les configurations d'ordinateurs que ce coût de recherche soit considérable et non négligeable par rapport au coût de construction de l'abr.

4.2.1 Résultats en moyenne sur la profondeur d'insertion

Le coût moyen d'une recherche vaine est ainsi égal à la profondeur moyenne d'insertion

$$\mathbb{E}(D_{n+1}) = \frac{1}{n+1}\mathbb{E}(E_n)$$

et d'après le lemme 3

$$\mathbb{E}(D_{n+1}) = \frac{1}{n+1}(\mathbb{E}(I_n) + 2n)$$

soit encore, par les évaluations ci-dessus,

$$\mathbb{E}(D_{n+1}) = 2(H_{n+1} - 1)$$

ce qui montre que le coût moyen de recherche est d'ordre de grandeur $2\log n$. On retrouve bien entendu le résultat obtenu par la méthode de séries génératrices de la section 3.4.

4.2.2 Résultats p.s. sur la longueur de cheminement externe

Rappelons que

$$E_n \;=\; \sum_{u \text{ nœud ext. de } T_n} |\, u \,|$$

soit encore

$$E_n = \sum_{k \geq 0} k\; U_k(n)$$

de sorte que

$$E_n = W_n'(1)$$

où W_n est le polynôme de niveau.

Comme la dérivée d'une martingale est encore une martingale, il suffit de calculer $M_n'(z)$ pour $z \in \mathbb{R}$, pour obtenir que

$$M_n'(1) = \frac{W_n'(1)}{n+1} - 2(H_{n+1} - 1)$$

est une martingale, soit le

Théorème 6 *(Regnier, 1989)*
$\frac{1}{n+1}(E_n - \mathbb{E}(E_n))$ *est une* $\mathcal{F}_n$*-martingale.*

On a, bien entendu, des résultats analogues pour la longueur de cheminement interne I_n.

References

[1] J. D. Biggins. (1997). How fast does a general branching random walk spread?. Classical and modern branching processes. IMA Vol. Math. Appl. **84** 19–39.

[2] B. Chauvin, M. Drmota and J. Jabbour-Hattab. (2001) The profile of binary search trees. to appear in Annals of Applied Probability

[3] L. Devroye. (1987). Branching processes in the analysis of the height of trees. Acta Inf. **24** 277–298.

[4] L. Devroye and B. Reed. (1995) On the variance of the height of random binary search trees, SIAM J. Comput. **24**, 1157-1162.

[5] M. Drmota. (2000) The Variance of the Height of Digital Search Trees, Acta Inf., submitted.

[6] J. Jabbour-Hattab. (2001). Martingales and large deviations for the binary search trees. Random Structures and Algorithms. **19**, 2, 112-127.

[7] H. M. Mahmoud. (1992) Evolution of Random Search Trees. Wiley.