# Random tree models in the analysis of algorithms

Philippe Flajolet

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.(1) 39 63 55 11

Rapports de Recherche

N° 729

# RANDOM TREE MODELS IN THE ANALYSIS OF ALGORITHMS

Philippe FLAJOLET

OCTOBRE 1987

# RANDOM TREE MODELS IN THE ANALYSIS OF ALGORITHMS

PHILIPPE FLAJOLET†

**Abstract.** *We present three classes of random tree models that occur in the average case analysis of a variety of computer algorithms including symbolic manipulation algorithms, compiling, comparison based searching and sorting, digital retrieval techniques, file systems and communication protocols. Each model carries a coherent set of algebraic and analytic techniques, which we illustrate by reviewing a few characteristic examples.*

# MODÈLES D'ARBRES ALÉATOIRES EN ANALYSE D'ALGORITHMES

**Résumé.** Nous decrivons ici trois classes de modèles d'arbres aléatoires qui se présentent naturellement dans l'analyse d'une grande variété d'algorithmes informatiques: systèmes de manipulation symbolique, compilation, recherche et tris par comparaisons, méthodes digitales, gestion de fichiers et protocoles de communication. Chaque modèle est lié à une ensemble cohérent de techniques algébriques et analytiques qui sont illustrées par quelques exemples caractéristiques.

† Invited lecture for the PERFORMANCE'87 Conférence, Brussels, December 7–9, 1987. Proceedings to be published by North Holland Pub. Co. (1987).

# RANDOM TREE MODELS IN THE ANALYSIS OF ALGORITHMS

PHILIPPE FLAJOLET
INRIA, Rocquencourt
78150 Le Chesnay (France)

**Abstract.** *We present three classes of random tree models that occur in the average case analysis of a variety of computer algorithms including symbolic manipulation algorithms, compiling, comparison based searching and sorting, digital retrieval techniques, file systems and communication protocols. Each model carries a coherent set of algebraic and analytic techniques, which we illustrate by reviewing a few characteristic examples.*

*ūrdhvamūlam adhaḥśākham aśvattham prāhur avyayam*
*chandāṁsi yasya parṇāni yas taṁ veda sa vedavit*
— Bhagavadgītā XV.1

## Introduction

In purely graph–theoretic terms, a *tree* is a connected acyclic (undirected) graph. The number of nodes in the tree is called the (total) *size* of the tree. As is familiar from data structures and basic algorithms [Knuth 1973], [Sedgewick 1983], a tree is a convenient way of representing a hierarchical object of some sort. As opposed to sequential structures (eg. list organizations), trees have the benefit of allowing dichotomic access to the information items of the structures they represent. Like many linked structures but unlike arrays, they are also easy to update.

Trees usually considered in computer science are trees in the general sense, but with some additional structure: (i) A node is individuated as the *root* of the tree. (ii) Subtrees attached to a node are ordered between themselves. This is equivalent to specifying a tree together with its plane embedding, and distinguishing subtrees dangling from a node with a left-to-right order. In this way, the general class of (rooted plane) trees $\mathcal{G}$ has a simple recursive specification: A single node is a tree; a new tree is obtained by appending a root to a sequence of trees already in $\mathcal{G}$. A particularly important subclass is the class $\mathcal{B}$ of binary trees in which every node has (out) degree 0 or 2.

Trees are also normally a structure superimposed on existing information items: records or "keys", operation symbols, program constructs, functional symbols etc. In other words, they are *labelled* trees. An algorithm may either use a tree as a direct representation of its input (this is the case for expression trees, terms), as an intermediate *data structure* whose design is meant to obtain fast processing of the data by allowing dynamic dichotomic search (a binary search tree or a digital "trie" are of this type). A *data model* or *input model* specifies a probability distribution over trees of size $n$. The simplest model

conceptually is the uniform model in which all trees in $\mathcal{G}$ or $\mathcal{B}$ of size $n$ are taken with equal probabilities.

There is an alternative, sometimes more convenient, way of viewing random tree models as *splitting processes* determined by splitting probabilities, which we explain in the binary case. A group $G$ (with $|G| = n$) is split recursively into two subgroups $L$ (for left) and $R$ (for right), where $|L| + |R| + \Delta = |G|$. There $\Delta$ is 1 or 0 depending on whether internal nodes contain informations (retain elements of $G$) or not. The process is fully characterized by its splitting probabilities

$$\pi_{n,k} = \Pr\{|L| = k \ / \ |G| = n\}$$

and a "termination rule", which is usually $|G| \leq 1$. The three basic models we consider in this paper are:

1. The *uniform (binary) model*, where all binary trees of size $n$ are taken equally likely. It has $\Delta = 1$, and will be seen to correspond to the splitting probabilities:

$$\pi_{n,k} = \frac{b_k b_{n-1-k}}{b_n} \qquad \text{where} \qquad b_m = \frac{1}{m+1}\binom{2m}{m}. \tag{1}$$

2. The *binary search tree model* has $\Delta = 1$ and

$$\pi_{n,k} = \frac{1}{n}. \tag{2}$$

3. The *digital trie model* has $\Delta = 0$ and splitting probabilities given by

$$\pi_{n,k} = \frac{1}{2^n}\binom{n}{k}. \tag{3}$$

The next three sections will explain the relevance of these models to applications, discuss a few related models, and provide typical examples of analytic resolution techniques. Detailed references and fairly comprehensive expositions of our subject appear in [Flajolet 1985] and [Vitter, Flajolet 1987].

## 1. Uniform Models and Variants

A formal (well–parenthetized) expression like $(x + \log y)/\sqrt{y * \log x}$ is naturally representable by a tree with a binary root (labelled with "/"), and two root subtrees corresponding to the left and right operands, and recursively constructible in a similar way. That expression can be represented as a term appearing in prefix form as $/ + x \log y \sqrt{} * y \log x$, and in Lisp notation it will be a "list" (/ (+ $x$ (log $y$)) ($\sqrt{}$ (* $y$ (log $x$)))), also internally represented as a (binary) tree. Many algorithms operate on such formal expressions. In its code generation phase, a compiler will need to generate low level instructions corresponding to the evaluation of the expression, and assign registers. A symbolic algebra system may have to apply to this expression a transformation like $\frac{\partial}{\partial x}$. Theorem proving applications and inference systems will also typically perform transformations on terms of a similar form.

Models in this section aim at analyzing, in the average case, algorithms of this sort, as a function of their input size. If $\Omega$ is a fixed set of functional symbols, the set of terms constructible on $\Omega$, $\mathcal{T} = \mathcal{T}[\Omega]$, is defined in an obvious way. The simplest input model consists in analyzing algorithms under the assumption that all trees in $\mathcal{T}$ with size $n$ are

taken equally likely. It is called the *uniform* model or *combinatorial* model (relative to $\Omega$ or $\mathcal{T}[\Omega]$), and sometimes also the "static" model, in contrast to the "dynamic" binary search tree model of the next section.

A more sophisticated model may assign a weight $w[f]$ to any individual symbol $f \in \Omega$, and we may wish to reflect in this way the fact that one symbol is twice as frequent as another one. If $t \in \mathcal{T}$ of size $n$ comprises symbols $f_1, f_2, \cdots, f_n$, then weights are extended multiplicatively and we take $w[t] = w[f_1]w[f_2]\cdots w[f_n]$. If $W_n$ designates the normalizing constant $W_n = \sum_{|t|=n} w[t]$, tree $t$ will be taken with probability $w[t]/W_n$. Such a model is called a *weighted model*. Weighted models should be expected to give more accurate predictions than uniform models. If each individual weight satisfies $w[f] = 1$, we get back the combinatorial (uniform) model. If $w[.]$ is a probability distribution over $\Omega$, then the weighted model resembles a branching process conditioned upon the size of the resulting tree, a fact that will be discussed later.

It turns out, fortunately, that the analytic techniques needed for uniform and weighted models are entirely similar. Thus, to simplify the discussion, we shall mostly concentrate on uniform models. The binary case leads to a model that can in many cases be solved exactly using generating function techniques. In all generality, however, complex analysis techniques are required and they lead to asymptotic expansions for expectations of parameters (random variables) of interest.

PROBLEM 1. *Determine the number of binary trees with $m$ binary nodes (hence $m+1$ external nodes and total size $n = 2m+1$).*

Let $B_n$ be the number of trees with total size $n$. We have $B_0 = 0$, $B_1 = 1$ and, by looking at all possibilities,

$$B_n = \sum_{l+r+1=n} B_l B_r. \tag{1}$$

The convolution equation is solved by introducing the ordinary generating function (OGF) $B(z) = \sum_{n\geq 0} B_n z^n$ which satisfies, by (1),

$$B(z) = z + zB^2(z), \tag{2}$$

and solving the quadratic equation, we find

$$B(z) = \frac{1-\sqrt{1-4z^2}}{2z}. \tag{3}$$

A standard Taylor expansion of $(1+x)^{1/2}$ shows that

$$B_{2m+1} = b_m \qquad \text{where} \qquad b_m = \frac{1}{m+1}\binom{2m}{m}. \tag{4}$$

Asymptotically, Stirling's formula yields

$$b_m \sim \frac{4^m}{\sqrt{\pi m^3}}. \tag{5}$$

The numbers $b_m$ are known as the Catalan numbers. They first occurred in works of Euler and Segner (around 1753!) and are of fundamental importance in combinatorial analysis. ∎

Looking back at our previous equations, we see that the $B_n$ are first defined by a convolution recurrence that reflects the recursive definition of trees. Introducing generating

functions shows that $B(z)$ is an algebraic function, with singularities of a square-root type. Finally, asymptotic analysis shows that $B_n \sim KA^n n^{-3/2}$. We are going to see how to extend these observations to the case of an arbitrary function symbol set $\Omega$.

PROBLEM 2. *Determine the number of trees of size $n$ in a family $\mathcal{T}$ defined by $\Omega$.*

Let $\phi_k$ denote the number of functional symbols in $\Omega$ that have arity equal to $k$, and introduce the *structure polynomial* $\phi(u) = \sum_k \phi_k u^k$. Let $T_n$ be the number of trees of size $n$. There is a convolution equation

$$T_n = \sum_{k\geq 1} \left[ \phi_k \sum_{n_1+\cdots+n_k+1=n} T_{n_1} T_{n_2} \cdots T_{n_k} \right], \tag{6}$$

which shows that the OGF $T(z)$ of the $T_n$ is defined implicitly by

$$T(z) = z\phi(T(z)). \tag{7}$$

Equation (7) shows that $T(z)$ is an *algebraic function*. In general, it cannot be solved in closed form. However, $T(z)$ is a solution of an equation $P(z, T(z)) = 0$ where $P(z, y) = y - z\phi(y)$. By the implicit function theorem, that equation has a solution that is locally analytic around $z = \varsigma$ with $T(z) = \tau$ provided $P'_y(\varsigma, \tau) \neq 0$, the dependence between $z$ and $T(z)$ being there locally linear:

$$P'_z(\varsigma, \tau)(z - \varsigma) + P'_y(T(z) - \tau) \sim 0.$$

From that observation, we find the dominant singularity $\rho$ of $T(z)$ as $\rho = \tau/\phi(\tau)$, where $\tau$ is the smallest positive solution of the equation $\phi(\tau) - \tau\phi'(\tau) = 0$. Furthermore around $(z, T) = (\rho, \tau)$ the dependence between $z$ and $T$ is now *locally* quadratic. Hence

$$(z - \rho) \sim C \cdot (T(z) - \tau)^2. \tag{8}$$

More precisely, a full expansion can be obtained (this is the classical Puiseux expansion in fractional powers of an algebraic function around a singularity), and we find that $T(z)$ admits the expansion

$$T(z) = H\left(\sqrt{1 - \frac{z}{\rho}}\right) \qquad \text{where} \qquad H(u) = H_0 + H_1 u + H_2 u^2 + \cdots. \tag{9}$$

In particular around $z = \rho$, $T(z)$ is of the form $H_0 + H_1\sqrt{(.)}$.

The Darboux method in asymptotic analysis [Henrici 1977] guarantees that, for a function with only algebraic singularities on its circle of convergence, a singular asymptotic expansion (9) of the function can be translated term by term into an asymptotic expansion of the coefficients. The coefficients of $\sqrt{1 - x}$ have a known asymptotic form (see Problem 1), hence

$$T_n \sim K\rho^{-n} n^{-3/2} \qquad \text{with} \qquad K = \sqrt{\frac{\phi(\tau)}{2\pi\phi''(\tau)}}. \tag{10}$$

The above derivation is due to Meir and Moon [1978]. It relies on classical singular expansions of algebraic functions, and also on Darboux's theorem whose proof is itself based on Cauchy's formula

$$T_n = \frac{1}{2i\pi} \oint T(z) \frac{dz}{z^{n+1}}. \tag{11}$$

The use of this type of technique in graphical enumerations goes back to the important paper of Pólya [1937]. ■

PROBLEM 3. *Determine the expectations, as a function of n, of inductive valuations over tree structures.*

There is an important class of tree parameters (random variables, "valuations") that covers many elementary statistics over trees. Let $u[t]$, $v[t]$, $w[t]$ be functions from trees to real numbers. To a valuation $u[t]$, we associate the generating function

$$U(z) = \sum_{t \in \mathcal{T}} u[t] z^{|t|}$$

so that the coefficient $U_n = [z^n]U(z)$ is the cumulated value of $u[.]$ over all trees of size $n$, and $\bar{U}_n = U_n/T_n$ is the corresponding expected value. Then, from convolution equations, we find "translation rules" from valuations to generating functions. For instance, when $\mathcal{T} = \mathcal{B}$, the class of binary trees, we have:

$$\begin{array}{lll} u[t] = v[t] + w[t] & \Longrightarrow & U(z) = V(z) + W(z) \\ u[t] = v[t_{\text{left}}] \cdot w[t_{\text{right}}] & \Longrightarrow & U(z) = zV(z)W(z). \end{array} \tag{12}$$

In particular, if $u[t]$ is defined inductively over subtrees in terms of a simpler valuation $v[t]$, we derive from (12):

$$u[t] = u[t_{\text{left}}] + u[t_{\text{right}}] + v[t] \quad \Longrightarrow \quad U(z) = \frac{1}{\sqrt{1-4z^2}} V(z). \tag{13}$$

For instance, path length is defined inductively over subtrees from the size function $v[t] = |t|$, and we find

$$U_{2n+1} = 4^n - \frac{3n+1}{n+1}\binom{2n}{n} \tag{14}.$$

In the same way, there is an extension to families $\mathcal{T} = \mathcal{T}[\Omega]$, for which the relation between $U(z)$ and $V(z)$ becomes

$$U(z) = \frac{1}{1 - z\phi'(T(z))} V(z).$$

A typical example of application [Steyaert, Flajolet 1983] is to counting occurrences of a fixed pattern tree $P$ inside trees in $\mathcal{T}$: The valuation "number-of-occurrences" is defined inductively from the valuation "root-occurrence", itself defined inductively over subtrees. In this manner, pattern matching algorithms on term trees can be analyzed. Another application [Flajolet, Steyaert 1987] is to symbolic differentiation whose cost is decomposable over subtrees and found to be always $O(n^{3/2})$. ■

PROBLEM 4. *Determine the expected height of a random binary tree with n internal nodes.*

This problem lies analytically deeper than the previous ones. Though height of a tree can be defined in terms of the heights of root subtrees, the definition involves a *max* operation that is not amenable to the treatment given in the previous section (Eq (12), (13)). From a computational point of view, the height parameter is of interest as it represents the maximum stack size necessary for a recursive traversal of the tree.

In this problem, we take the size of a binary tree to be the number of its internal nodes. Height is defined as the length of the longest branch from the root to an external node (or leaf), with the convention that a tree of size 0 has height 1. We let $b_{h,n}$ be the number of

binary trees with height at most $h$ and size $n$, with OGF $b_h(z) = \sum_n b_{h,n} z^n$. The problem is to determine the expected height $\bar{H}_n = H_n/b_n$, where $H_n = \sum_{|t|=n} \text{height}[t]$ satisfies

$$H_n = \sum_{h \geq 0} [b_n - b_{h,n}]. \tag{15}$$

The difficulty comes from the fact that the $b_{h,n}$ satisfy non-linear "history" recurrences for which no closed form solution is likely to exist. Our presentation follows [Flajolet, Odlyzko 1983] (see [Brown, Shubert 1984] and [Kolchin 1986] for related treatments).

From basic combinatorial decompositions, we derive the recurrence

$$b_0(z) = 1; \qquad b_{h+1}(z) = 1 + z(b_h(z))^2, \tag{16a}$$

and the OGF of $H_n$ is related to the $b_h(z)$ via

$$H(z) = \sum_{h \geq 0} [b(z) - b_h(z)], \tag{16b}$$

with $b(z)$ the fixed point of recurrence (16a) and also the OGF of the Catalan numbers:

$$b(z) = \sum_{n \geq 0} b_n z^n = \frac{1 - \sqrt{1-4z}}{2z}.$$

Function $H(z)$ is expected to be singular at $z = 1/4$ since $H_n \leq n b_n$ and by (5), $b_n$ grows roughly like $4^n$. The attack consists in establishing the behaviour of $H(z)$ in a neighbourhood of $z = 1/4$ and analysis will reveal that

$$H(z) \sim C \log \frac{1}{1-4z} \qquad \text{as} \qquad z \to 1/4. \tag{17a}$$

The estimate of $H_n$ will be completed if we can "transfer" the asymptotic equivalence (17a) into an asymptotic equivalence for coefficients:

$$H_n \sim C \frac{4^n}{n} \qquad \text{as} \qquad n \to \infty, \tag{17b}$$

and, if this is granted, we get for the expected height $\bar{H}_n \sim C\sqrt{\pi n}$.

Darboux's method is one way of ensuring the transition from (17a) to (17b), but it requires differentiability conditions on error terms that are rather stringent for this application. Tauberian methods are not applicable since they apply to functions that are large around their singularity (while the remainder term implicit in (17a) is small) and, in addition, require side conditions that would be hard to establish on error terms. Thus, we take another route, suggested by [Odlyzko 1982]. It consists in establishing approximation (17a) in an area of the complex plane that extends *beyond* the circle of convergence $|z| = 1/4$ of $H(z)$, using Cauchy's formula (11) with a contour of integration that leaves the singularity "at an angle" from the circle. That method is quite general and is rapidly finding applications in combinatorial enumerations [Flajolet, Odlyzko 1987].

What is required is thus to establish (17a), on the basis of the recurrence (16a-b). We observe that when $|z| < 1/4$, the $b_h(z)$ converge geometrically to the fixed point $b(z)$. When $|z|$ is large, because of repeated squaring, the $b_h(z)$ grow doubly exponentially, and for instance, at $z = 1$, their values are 1, 2, 5, 26, 676, 456976 etc. We are thus facing what

may be called a "singular iteration problem" which consists in obtaining approximations for recurrence (16a) when $z$ is in a region near a singularity of the fixed point, where the recurrence is just between geometric convergence and double exponential divergence.

The normalized "écarts" $e_h(z) = (b(z) - b_h(z))/(2b(z))$ satisfy the recurrence

$$e_h(z) = (1 - \sqrt{1-4z})e_h(z)(1 - e_h(z)) \tag{18}$$

and studying them at $z = 1/4$ illustrates the analytic technique used. There, setting $f_h = e_h(1/4)$, we find

$$f_{h+1} = f_h(1 - f_h) \qquad \text{with} \qquad f_0 = \frac{1}{2}. \tag{19}$$

Iteration (19) is typical of iteration of a function $g(x)$ (here $g(x) = x(1-x)$) around a fixed point $L = g(L)$ when $g'(L) = 1$. The essential "trick" is to compare iteration (19) to iteration of a linear fractional transformation. More precisely, we start from (19), and take *inverses* to get

$$\frac{1}{f_{h+1}} = \frac{1}{f_h} \cdot \frac{1}{1 - f_h} = \frac{1}{f_h} + 1 + f_h + f_h^2 + \cdots. \tag{20}$$

Thus, as a rough approximation $f_{h+1}^{-1} \approx 1 + f_h^{-1}$ so that we expect $f_h^{-1} \approx h$. This can be justified rigorously, and indeed, one can show that

$$f_h \sim \frac{1}{h + \log h + O(1)}.$$

A similar manipulation on the more general recurrence (18) provides an approximation for $e_h(z)$, in the form

$$b(z) - b_h(z) \approx 4\epsilon \frac{(1-\epsilon)^h}{1 - (1-\epsilon)^h} \qquad \text{with} \qquad \epsilon = \epsilon(z) = \sqrt{1-4z}. \tag{21}$$

Equation (21) is the main approximation lemma. Its validity region can be established at some effort. Basically, we are justified in using it inside the definition (16b) of $H(z)$:

$$H(z) \sim 4 \sum_h \epsilon \frac{(1-\epsilon)^h}{1 - (1-\epsilon)^h} \sim 4 \sum_h \frac{(1-\epsilon)^h}{h} = 2 \log \frac{1}{1-4z}, \tag{22}$$

as $z \to 1/4$, ie. $\epsilon \to 0$. We have thus found the singular expansion (17a) and the proof can be concluded using Cauchy's formula. We find in this way for the expected height of a binary tree with $n$ internal nodes:

$$\bar{H}_n = 2\sqrt{\pi n} + O(n^{0.26}). \tag{23}$$

The same treatment applies *mutatis mutandis* to families $\mathcal{T}[\Omega]$. The singular iteration problem we encountered in the complex plane is related to Mandelbrot and Julia fractal sets. In general, no explicit form is likely to exist for such non linear iterations. There is however an interesting exception, namely the case of trees where all node degrees are allowed. It was studied by [De Bruijn, Knuth, Rice 1972] and corresponds to a structure function $\phi(u) = (1-u)^{-1}$. In that case, height statistics requires iteration of a linear fractional transformation, and explicit forms, related to continued fractions, are known.

An expression resembling the approximation (21) then appears as an *exact* formula for the OGF of trees with height at most $h$, so that the asymptotic analysis of height can be done with less analytic machinery.

Let us mention finally that there is some superficial resemblance (that can be justified in particular cases) between this problem and the height of random walks. Most notably, there is a limiting distribution for height whose density distribution involves a *theta* function. ■

The weighted model can be subjected to the same analytic treatment. If the weights $w[f]$ with $f \in \Omega$ are a probability distribution, it is then easy to see that the model is equivalent to a branching process conditioned upon the size $n$ of the resulting trees. A particularly interesting case is when the branching process is "critical" (the expected number of offsprings in a generation is 1). That condition corresponds to a natural combinatorial conservation condition on edges, and asymptotically each node type $f$ occurs with probability $w[f]$.

Statistical study conducted by Clarke [1977] on large Lisp data structures suggest that a branching process model is a reasonable approximation to reality, with parameters rather independent of specific applications. This adds some practical justification for the study of models considered in this section when applied to terms, expression trees and Lisp data structures.

## 2. The Binary Search Tree Model

In this section, we consider the splitting model defined by Eq (1.2) that arises in the study of several data structures used for maintaining dynamically varying collections of data belonging to an ordered domain. The model underlying this random tree model is that of random permutations. It is practically justified when data items are produced independently according to a continuous probability distribution.

*Heap-ordered trees.* Heap-ordered trees are binary trees whose binary (internal) nodes are labelled with distinct integers, in such a way that labels along any branch from the root of the tree, the labels form an increasing sequence. Given a permutation $\sigma = \sigma_1\sigma_2\cdots\sigma_n$, the associated heap-ordered tree $\mathrm{HOT}(\sigma)$ is defined by a simple recursive rule. Let $j$ be the position of the smallest element in $\sigma$; then

$$\mathrm{HOT}(\sigma) = \langle \mathrm{HOT}(\sigma_1\sigma_2\cdots\sigma_{j-1})\,;\,\sigma_j\,;\,\mathrm{HOT}(\sigma_{j+1}\sigma_{j+2}\cdots\sigma_n)\rangle. \tag{1}$$

Conversely, given a HOT, a permutation is obtained by reading the labels in left-to-right (infix) order.

Thus, heap-ordered trees constitute a bijective representation of permutations. The uniform distribution over the symmetric group of rank $n$ (where each permutation is taken with probability $1/n!$) induces a probability distribution over the set $\mathcal{B}_n$ of binary trees with $n$ internal nodes, when one "forgets" the labels. (Here again, it is convenient to define the size of a binary tree as the number of its internal nodes). That distribution is certainly non uniform since in general $b_n$ does not divide $n!$. The probability that the left subtree, in a tree of size $n$, has size $k$ is equal to $1/n$ independently of the value of $k$, $0 \le k \le n-1$, since in a random permutation, the minimal element can occur in any position with equal probability. This justifies consideration of the splitting model (0.2) of the introduction in which

$$\pi_{n,k} = \frac{1}{n}. \tag{2}$$

HOT's are only one amongst a whole class of data structures that are useful for implementing priority queues: The "find-minimum" operation is a direct access to the

root of the tree and has thus cost $O(1)$. Operations of "delete-minimum" and "insert" can also be performed efficiently. In [Gonnet 1984] and [Vuillemin 1980], one can find a discussion of priority queue implementations and data structures related to heap-ordered trees: heaps, pagodas, binomial queues, leftist trees etc.

*Binary search trees.* Binary search trees (BST's) and their "balanced variants" (AVL trees, 2-3 trees, B-trees, dichromatic trees) permit to implement the full set of so-called dictionary operations, namely insert, delete, query [Knuth 1973], [Sedgewick 1983], [Gonnet 1984]. It $\tau = \tau_1\tau_2\cdots\tau_n$ is a sequence of elements (a "file"), then the BST associated to $\tau$ has $\tau_1$, the first arrived element, at its root. If $\tau_<$ and $\tau_>$ are the subsequences of $\tau_2\tau_3\cdots\tau_n$ formed with elements smaller, resp. larger, than $\tau_1$, then $\mathrm{BST}(\tau)$ is defined recursively by the rule

$$\mathrm{BST}(\tau_1\cdots\tau_n) = \langle \mathrm{BST}(\tau_<)\,;\,\tau_1\,;\,\mathrm{BST}(\tau_>)\rangle. \tag{3}$$

Thus if the tree labels are read in left-to-right order, we obtain the sorted file.

It turns out, by an equivalence principle of Burge and Françon (see [Vuillemin 1980]), that there is a strong connection between HOT's and BST's: If $\sigma$ and $\sigma^{-1}$ are inverse permutations, then

$$\mathrm{HOT}(\sigma) \equiv_{\mathrm{shape}} \mathrm{BST}(\sigma^{-1}) \tag{4}$$

where $t \equiv_{\mathrm{shape}} u$ means that the unlabelled trees corresponding to $t$ and $u$ are identical. Thus the random tree model induced by (2) also applies to BST's and is called the *binary search tree model.* Most parameters that determine the cost of BST or HOT implementations are natural structural parameters (valuations) of trees. So one needs methods to determine their expected values, as a function of $n$, under the BST model. In addition, characteristic parameters of comparison based sorting routines, most notably Quicksort, have direct counterparts under the BST model.

Counting problems like in Section 1 have now become trivial, since the underlying statistics is the number $n!$ of permutations. Thus, we start directly with an analogue of Problem 3.

PROBLEM 5. *Determine, as a function of $n$, the expectations of inductive valuations under the BST model.*

The approach is here very similar to that of Problem 3. Only the generating functions used and the operators translating valuations are different. If $u[t]$ is a tree valuation, we let $U_n$ denote its expectation under the BST model of degree $n$, and introduce the OGF of expectations

$$U(z) = \sum_{n\geq 0} U_n z^n.$$

We have the following analogue of Eq (12):

$$\begin{array}{lcl} u[t] = v[t] + w[t] & \Longrightarrow & U(z) = V(z) + W(z) \\ u[t] = v[t_{\mathrm{left}}]\cdot w[t_{\mathrm{right}}] & \Longrightarrow & U(z) = \int_0^z V(x)W(x)\,dx. \end{array} \tag{5}$$

The first equation only expresses linearity of expectations. The second comes directly by taking generating functions in

$$U_n = \frac{1}{n}\sum_{k=0}^{n-1} U_k V_{n-k},$$

this equation being a direct consequence of (2). If $u[t]$ is defined inductively over subtrees from $v[t]$,

$$u[t] = u[t_{\text{left}}] + u[t_{\text{right}}] + v[t], \tag{6a}$$

then, by (5), the corresponding OGFs are related by

$$U(z) = 2\int_0^z U(x)\frac{dx}{1-x} + V(z) \tag{6b}$$

and the corresponding differential equation is readily solved by the "variation-of-parameter" method, so that

$$U(z) = \frac{V(0)}{(1-z)^2} + \frac{1}{(1-z)^2}\int_0^z (1-x)^2 V'(x)\,dx. \tag{6c}$$

The study of inductive valuations now takes us into the realm of integral and differential operators, while uniform models only implied algebraic operators. An interesting application is internal path length, for which we find, through an application of (6c),

$$U_n = 2(n+1)H_{n+1} - 2n - 2 \sim 2n\log n + O(n),$$

with $H_n = 1^{-1} + 2^{-1} + \cdots + n^{-1}$, a harmonic number. This quantity represents the expected comparison cost of treesort, the procedure that sorts a file by constructing a binary search tree. Like quicksort, that can be subjected to a similar analysis, it has complexity $\approx 1.3862 n \log_2 n$ and is thus about 40% off from optimum sorting. ■

The expected height of a binary tree with $n$ (internal) nodes is $\sim 4.31107 \log n$. This remarkable result is due to Devroye [1986] who established it using the theory of branching random walks. Though basic generating function equations can be set up like for Problem 4, with integrals replacing products, it is still an open problem to find a purely analytic derivation that should involve a treatment of singular Picard approximants to a non linear ODE around a spontaneous singularity. So we shall turn again to a study of more complex inductive valuations in the style of Problem 5. These are related to the analysis of partial match retrieval of multidimensional data, following [Flajolet, Puech 1986].

PROBLEM 6. *Determine the cost of finding a record in* $\mathbf{R}^2$ *specified by one component, in a 2-dimensional search tree.*

An extension of binary search trees, called $k$-dimensional ($k$-d) trees stores $k$ dimensional points at nodes of a binary tree, and uses, at level $\ell$ in the tree, component ($\ell \bmod k$) as a discriminating attribute in the style of binary search trees. The probability distribution induced over binary trees appears to coincide with that of the BST model. For instance, the number of nodes traversed when searching an element in the tree has the same distribution in dimension $k$ and in dimension 1. A *partial match query* has only $s$ out of the $k$ attributes that are specified. At levels in the tree where the discriminating attribute is specified, search proceeds in only one subtree; at other levels, both subtrees have to be explored. In this way, the analysis of partial match retrieval necessitates analyzing $k$ different valuations that are cyclically related.

We shall first briefly explain here the mathematics involved in the case $k = 2$ and $s = 1$. There are two patterns of partial match queries, $*S$ and $S*$, where "$S$" means a specified attribute and "$*$" means unspecified. We let $C_{P,n}$ denote the expected cost of a match with pattern $P$ and file size $n$, and introduce the generating functions $C_P(z) = \sum_n C_{P,n} z^n$, $D_P(z) = \sum_n (n+1) C_{P,n} z^n$. A development of the same spirit as that of Problem 5 shows

that $D_{S*}(z) = y_1(z)$ where $y_1(z)$ is a component of the differential system

$$\frac{d}{dz}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 & 2(1-z)^{-1} & -z^{-2}(1-z)^{-1} \\ 2(1-z)^{-1} & z^{-1}(1-z)^{-1} & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} + \begin{pmatrix} 2(1-z)^{-3} \\ (1-z)^{-3} \\ 0 \end{pmatrix}. \tag{7}$$

(We also have $D_{*S}(z) = y_2(z) = y_3'(z)$.) Various methods would make it possible to reduce the system to a unique differential equation of order 3, but it is just as simple to continue with the original system. System (7) can be put in the form

$$\frac{d}{dz}\mathbf{y}(z) = \frac{1}{1-z}\mathbf{A}(z) \cdot \mathbf{y}(z) + \mathbf{b}(z) \tag{8}$$

with $\mathbf{y}$ the column vector of the $y_i$ etc. The interesting property is that $\mathbf{A}(z)$ is analytic at $z = 1$. In the classical theory of differential systems, we say that $z = 1$ is a *regular singular point* [Henrici 1977], and the solutions should have algebraico-logarithmic singularities at this point.

Again, no closed form expression is likely to exist for the solutions but asymptotic expansions around the singularity $z = 1$ can be obtained. First, one examines the homogeneous version of system (8), namely

$$\frac{d}{dz}\mathbf{y}(z) = \frac{1}{1-z}\mathbf{A}(z) \cdot \mathbf{y}(z). \tag{9}$$

If $\mathbf{A}(z)$ is constant, $\mathbf{A}(z) = \mathbf{A}(1)$, the system is called an Euler equation. Then by diagonalizing $\mathbf{A}(1)$, one finds explicit solutions that are combinations of functions of the form $(1-z)^\lambda$ with $\lambda$ an eigenvalue of $\mathbf{A}(1)$. The characteristic polynomial of $\mathbf{A}(1)$ equated to zero gives the *indicial equation*, and its solutions determine the characteristic exponents that appear in the solution of system (9) even when $\mathbf{A}(z)$ is not constant.

In this way, one can determine full asymptotic expansions around $z = 1$ of solutions to the homogeneous system (9). Solutions to the inhomogeneous system (8) can be generated either by a method of indeterminate coefficients or by a matrix version of the variation-of-constant method. In our case, one finds the indicial equation in the form

$$\lambda(\lambda + 1) - 4 = 0 \tag{10}$$

so that

$$D_{S*}(z) \sim C(1-z)^\lambda \qquad \text{with} \qquad \lambda = -\frac{1+\sqrt{17}}{2}. \tag{11}$$

The final stage can use either Darboux's method or, more conveniently, transfer theorems briefly discussed in relation to Problem 4. This enables us to translate the asymptotic expansion of generating function $D$ around its singularity into a corresponding asymptotic form for coefficients, and get the final result

$$C_{S*,n} \sim K \cdot n^\alpha \qquad \text{with} \qquad \alpha = \frac{\sqrt{17}-3}{2} \approx 0.57.$$

The same approach can be extended, at some effort, to the study of the general case of $k$-d trees of arbitrary dimensions. What is required is to extract the structure of the

matrix $\mathbf{A}(z)$ and especially of $\mathbf{A}(1)$ and its characteristic polynomial which is found to be of a simple form,

$$\chi(\lambda) = \lambda^s(1+\lambda)^{k-s} - 2^k,$$

a natural generalization of form (10). ■

The same approach has the advantage of extending to many *probabilistic divide-and-conquer recurrences* of the form

$$f_n = K\sum \pi_{n,k} f_k + g_n. \tag{12}$$

Most $\pi_{n,k}$ that are rational fractions in $n$ and $k$ will yield to this treatment. Others as well, as shown by the case of Quad-Trees (Flajolet, Gonnet, Puech, Robson, unpublished) for which

$$\pi_{n,k} = \frac{1}{n}[H_n - H_k] \qquad \text{where} \qquad H_m = 1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{m}.$$

## 3. The Digital Trie Model

Perhaps the simplest description of the random tree model corresponding to tries is in terms of splitting processes. A group $G$ is split recursively into two subgroups $G_0$ and $G_1$ by having independently each member of $G$ flip a coin. The process is halted as soon as a subgroup with cardinality $\leq b$ is obtained. An execution of the process is described by a binary tree in which internal nodes represent the splitting phases while external nodes contain groups of cardinality at most equal to $b$. Here we shall be primarily concerned with the threshold parameter $b$ having value 1.

*Tries.* The trie data structure is a tree representation of sets of binary strings. If a file $F$ is to be stored, it can be organized as a tree, called a *trie*, with the left and right subtrees associated to $F_0$ and $F_1$, where $F_j$ is the subset of strings starting with 0 and 1 respectively (stripped of their original bit). Thus, search in such a tree of a "key" $\beta$ is obtained by following a path guided by successive bits of $\beta$ in the tree trie($F$) until a leaf is found. In a similar way, if $b$ is the page capacity of a disk, the trie can be used as a "directory" to access the file on disk: When used in connection with hashing, this gives rise to Dynamic Hashing or Extendible Hashing [Larson 1978], [Fagin *et al* 1979].

*Protocols.* The tree communication protocol of Tsybakov-Mikhailov and Capetanakis manages collisions on a local area network in the following way: Stations are granted access to the channel upon arrival of a message. In case of a collision (two or more users attempt transmission simultaneously), we use the splitting process above to resolve their access conflict [Massey 1981], [Massey 1985].

These two implementations of the abstract splitting process, tries and protocols, require analysis. Path length in the trie represents the cost of searching all elements, and its expectation (divided by $n$, the group size) represents the expected cost of a positive search. Similarly, the session length in the protocol implementation corresponds to the total number of nodes in the associated tree. Thus, methods for estimating such inductive tree parameters are of special value. From analysis it results that the expected search time in a trie built on random binary strings is close to $\log_2 n$, and thus is close to a theoretical optimum. The "throughput" of the tree protocol is $\log 2 \pm 10^{-5}$.

PROBLEM 7. *Determine as a function of the group size $n$, the expectation of inductive tree parameters under the random trie model.*

For a fair coin, the splitting probabilities are given by

$$\pi_{n,k} = \frac{1}{2^n}\binom{n}{k},$$

as they are Bernoulli trial probabilities. The analysis method consists, as usual, of two parts: An easy translation from valuation specifications into (here exponential) generating functions of expected values. Asymptotic methods that introduce interesting Mellin transform techniques, originally a tool from analytic number theory. We can state rules that are analogues of Eq (1.12), (1.13) for random plane trees, and of (2.5), (2.6) for random binary search trees. It $u[t]$ is a tree parameter and $U_n$ is its expectation under the assumption that the tree is a trie randomly built from a group $G$ of size $n$, then we introduce the exponential generating function (EGF) of expectations

$$U(z) = \sum_{n\geq 0} U_n \frac{z^n}{n!}. \tag{1}$$

We find the basic translation rules

$$\begin{array}{lll} u[t] = v[t] + w[t] & \Longrightarrow & U(z) = V(z) + W(z) \\ u[t] = v[t_{\text{left}}] \cdot w[t_{\text{right}}] & \Longrightarrow & U(z) = V(\frac{z}{2}) \cdot W(\frac{z}{2}) \end{array} \tag{2}$$

The second part of the rule is directly inferred from the convolution equation resulting from (1) and the definition of the $\pi_{n,k}$,

$$U_n = \sum_{k\geq 0} \frac{1}{2^n}\binom{n}{k} V_k W_{n-k}.$$

In particular, if $u[.]$ is defined inductively from $v[.]$,

$$u[t] = u[t_{\text{left}}] + u[t_{\text{right}}] + v[t], \tag{3a}$$

then, from (2), we get a *difference equation*,

$$U(z) = 2e^{z/2}U(\frac{z}{2}) + V(z). \tag{3b}$$

There are two ways of solving (3b). The first method proceeds by iteration, and one gets the form

$$U(z) = \sum_{j\geq 0} 2^j \left[e^{z(1-2^{-j})} V(\frac{z}{2^j})\right]. \tag{4}$$

Extracting the coefficients of (4) permits to express the $U_n$ in terms of the $V_n$.

The alternative method introduces the *Poisson generating functions*

$$\hat{U}(z) = e^{-z}U(z), \qquad \hat{V}(z) = e^{-z}V(z) \tag{5}$$

that represent the expectations of $u[.]$ and $v[.]$ when the size is itself a random variable with a Poisson distribution of parameter $z$. Then (3b) becomes

$$\hat{U}(z) = 2\hat{U}(\frac{z}{2}) + \hat{V}(z), \tag{6}$$

which can be solved by the method of indeterminate coefficients (with $\hat{U}_n$ the coefficient of $z^n/n!$ in $\hat{U}(z)$):

$$\begin{aligned}\hat{U}_n &= 2^{1-n}\hat{U}_n + \hat{V}_n \\ &= \frac{\hat{V}_n}{1-2^{1-n}}.\end{aligned} \tag{7}$$

Finally, from (7), we can recover the $U_n$ from the $\hat{U}_n$ by a convolution, since $U(z) = e^z\hat{U}(z)$:

$$U_n = \sum_{k=0}^{n} \binom{n}{k} \hat{U}_k = \sum_{k=0}^{n} \binom{n}{k} \frac{\hat{V}_k}{1-2^{1-k}}. \tag{8}$$

For instance, when we take $v[t]$ to be 1 if the size of (the group associated to) $t$ is $\geq 2$ and 0 otherwise, $v[t]$ is the number of internal nodes in trie $t$. We have then $V(z) = e^z - 1 - z$ and by (4)

$$U(z) = \sum_{j\geq 0} 2^j \left[e^z - (1 + \frac{z}{2^j})e^{z(1-2^{-j})}\right] \tag{9}$$

and extracting coefficients, we find

$$U_n = \sum_{j\geq 0} 2^j \left[1 - (1-\frac{1}{2^j})^n - \frac{n}{2^j}(1-\frac{1}{2^j})^{n-1}\right]. \tag{10}$$

If we take the second route, we find $\hat{V}(z) = 1 - (1+z)e^{-z}$, so that $\hat{V}_n = (-1)^{n-1}(n-1)$ for $n \geq 2$, and (8) yields the alternative form

$$U_n = \sum_{k=2}^{n} \binom{n}{k} \frac{(-1)^{k-1}(k-1)}{1-2^{1-k}}. \tag{11}$$

Similarly, using $v[t] = |t|$ (ie. the size of the underlying group) corresponds to taking $V(z) = z(e^z - 1)$, and yields path length (taken over non empty external nodes), for which

$$U_n = n\sum_{j\geq 0} \left[1 - (1-\frac{1}{2^j})^{n-1}\right] = \sum_{k\geq 1} \frac{(-1)^{k-1}k}{1-2^{1-k}}. \tag{12}$$

This systematic chain makes it possible to analyze exactly a large number of parameters of random tries. ■

PROBLEM 8. *Analyze asymptotically the expectations of inductive tree parameters under the random trie model.*

The asymptotic analysis of expectations of trie parameters is not as innocuous as might seem at a first glance. The difficulty lies in the fact that sums like (10) or (11) do not appear to have a smooth growth decribable in terms of standard functions like $n$, $\log n$ etc. Instead non-trivial periodicity phenomena appear, and for instance, in the case of path length, $U_n/n$ is an asymptotically periodic function of $\log_2 n$. The main technique used is the Mellin transform (see eg [Flajolet, Régnier, Sedgewick 1985]) whose basic steps are:

1. Consider $U_n$ (or a closely related function) as a real function of a real variable. For instance, in the case of path length, we have $U_n = nF(n)$ where

$$F(x) = \sum_{j\geq 0} \left[1 - (1-\frac{1}{2^j})^x\right]. \tag{13}$$

We may also use approximate interpolation functions. Thus, using $(1-a)^n \approx \exp(-na)$, we can justify by elementary means that $U_n \sim nG(n)$ where

$$G(x) = \sum_{j\geq 0} \left[1 - e^{-x/2^j}\right]. \tag{14}$$

2. Compute the Mellin transform of the real function interpolating $U_n$ (the exact $F(x)$ or the approximate $G(x)$). The Mellin transform of a real function $f(x)$ is a function of the complex variable $s$ defined by

$$f^*(s) = \int_0^\infty f(x) x^{s-1}\, dx. \tag{15}$$

A sum like (13) or (14) is called a *harmonic sum*. It obeys the general pattern

$$f(x) = \sum_j \lambda_j g(\mu_j x) \tag{16}$$

and by analogy with Fourier analysis, we may consider the $\lambda_j$ to be amplitudes and the $\mu_j$ to be frequencies. The Mellin transform of a harmonic sum (16) is easily found to be

$$f^*(s) = \left(\sum_j \lambda_j \mu_j^{-s}\right) \cdot g^*(s), \tag{17}$$

which is thus decomposed as a product of the transform $g^*(s)$ of the base function and a Dirichlet series involving only frequencies and amplitudes. In the case of (14), we find

$$G^*(s) = -\frac{\Gamma(s)}{1-2^s}. \tag{18}$$

3. Finally, there is a well–recognized correspondence [Doetsch 1955] between singularities of a Mellin transform in a half-plane extending to the right of its definition strip and terms of the asymptotic expansion of the original function as $x \to \infty$. For instance, transform (18) is defined for $-1 < \Re(s) < 0$, and the double pole at $s = 0$ contributes the term

$$\log_2 x + \left(\frac{\gamma - 1}{\log 2} - \frac{1}{2}\right). \tag{19}$$

Complex poles of the Mellin transform further contribute fluctuating terms [Knuth 1973, p.131], [Flajolet, Régnier Sedgewick 1985] in the form of a Fourier series involving values of the Gamma function on the imaginary line. ■

Applying the algebraic techniques of Problem 7 and the asymptotic methods of Problem 8 constitutes a production chain (that could be automated using symbolic algebra systems) which applies to a large number of characteristics of tries. It is in this way that numerical results mentioned at the beginning of this section can be established.

We obtain that tries are an almost optimal data structures, that pages are about $\log 2 \approx 69\%$ full if dynamic hashing schemes are used. We also find that resolving a collision between $n$ users in the tree protocol requires on the average $\approx 2n/\log 2$ slots. Thus the throughput of the blocked access version of the protocol is close to $\log 2/2 \approx 34.65\%$. Analysis of the free access version can also be made along similar lines, though at a considerably greater effort (see [Flajolet, Jacquet 1987] for a brief survey).

## Conclusion

Each of the random tree model we have examined carries a coherent set of algebraic techniques by which parameter specifications are mapped to generating function equations, which are then solved if elementary enough. In a second phase, asymptotic methods are applied to extract useful information, and they normally rely on complex analysis. We have tried to illustrate this philosophy on three classes of problems:

1. Uniform tree models with algebraic equations and algebraic functions as solutions. Singularity analysis is the main asymptotic tool there.
2. Binary search trees. This is the realm of differential systems. Either they have elementary solutions or, otherwise, singularity analysis is applied at regular singular points.
3. Tries lead to difference equations and Mellin transforms.

## References

G. G. BROWN AND B. O. SHUBERT [1984]. "On Random Binary Trees", *Mathematics of Operations Research* **9**, 1984, 43–65.

N. G. DE BRUIJN, S. O. RICE AND D. E. KNUTH [1972]. "The Average Height of Planted Plane Trees" in *Graph Theory and Computing*, R. C. Read Ed., Academic Press, 1972, 15–22.

D. W. CLARKE [1977]. "An Empirical Study of List Structure in Lisp", *Comm. ACM* **20**, 1977, 78–87.

L. DEVROYE [1986]. "A Note on the Height of Binary Search Trees", *JACM* **33**, 1986, 489–498.

G. DOETSCH [1955]. *Handbuch der Laplace Transformation*, 3 Volumes, Birkhäuser Verlag, Basel, 1955.

R. FAGIN, J. NIEVERGELT, N. PIPPENGER AND H. R. STRONG [1979]. "Extendible Hashing, A Fast Access Method for Dynamic Files", *ACM Trans. on Database Systems* **4**, 1979, 315–344.

PH. FLAJOLET [1985]. "Mathematical Methods in the Analysis of Algorithms and Data Structures," INRIA, Research Report **400**, 1985. To appear in *A Graduate Course in Computer Science*, Computer Science Press, 1987.

PH. FLAJOLET AND PH. JACQUET [1987]. "Analytic Models for Tree Communication Protocols", INRIA Research Report **648**, 1987. To appear in *Flow Control of Congested Networks*, Springer NATO series.

P. FLAJOLET AND A. M. ODLYZKO [1982]. "The Average Height of Binary Trees and Other Simple Trees", *J. Computer and System Sciences* **25**, 1982, 171-213.

P. FLAJOLET AND A. M. ODLYZKO [1987]. "Singularity Analysis of Generating Functions", preprint, 1987.

P. FLAJOLET AND C. PUECH [1986]. "Partial Match Retrieval of Multidimensional Data", *JACM* **33**, 1986, 371–407.

P. FLAJOLET, M. RÉGNIER AND R. SEDGEWICK [1985]. "Some Uses of the Mellin Integral Transform in the Analysis of Algorithms", in *Combinatorics on Words*, Springer NATO ASI Series F, Volume 12, Berlin, 1985.

P. FLAJOLET AND J–M. STEYAERT [1987]. "A Complexity Calculus for Recursive Tree Algorithms", *Math. System Theory* **19**, 1987, 301–331.

G. H. GONNET [1984] *Handbook of Algorithms and Data Structures.* Addison-Wesley, Reading, 1984.

P. HENRICI [1977]. *Applied and Computational Complex Analysis.* Three Volumes. Wiley, New York, 1977.

P. JACQUET AND M. RÉGNIER [1987]. "Normal Limiting Distribution of the Size of Tries", in *Performance'87, Proceedings of 13th Int. Symp. on Computer Performance*, Brussels, 1987 (to appear).

D. E. KNUTH [1973]. *The Art of Computer Programming.* Volume 3: *Sorting and Searching.* Addison-Wesley, Reading, MA, 1973.

P. A. LARSON [1978]. "Dynamic Hashing", *BIT* **18**, 1978, 184–201.

J. MASSEY [1981]. "Collision Resolution Algorithms and Random Access Communication", in *Multi–User Communication Systems*, G. Longo Ed., CISM Courses and Lectures **235**, Springer, New–York 1981, 73–137.

J. MASSEY [1985]. *IEEE Trans. on Information Theory* **31**, Special Issue on Random Access Communication, J. Massey Ed., 1985.

A. MEIR AND J. W. MOON [1978]. "On the Altitude of Nodes in Random Trees," *Canadian Journal of Mathematics* **30**, 1978, 997–1015.

A. M. ODLYZKO [1982]. "Periodic Oscillations of Coefficients of Power Series that Satisfy Functional Equations", *Advances in Math.* **44**, 1982, 180-205.

G. POLYA [1937]. "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen", *Acta Mathematica* **68**, 1937, 145-254. Translated in: G. Polya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*, Springer, New York, 1987.

R. SEDGEWICK [1983]. *Algorithms.* Addison-Wesley, Reading, 1983.

J–M. STEYAERT AND PH. FLAJOLET [1983]. "Patterns and Pattern–Matching in Trees: An Analysis", *J. Comp. and System Sc.* **58**, 1983, 19–58.

J. VITTER AND PH. FLAJOLET [1987]. "Average Case Analysis of Algorithms and Data Structures", in *A Handbook of Theoretical Computer Science*, North Holland Pub. Comp., 1987, to appear.

J. VUILLEMIN [1980]. "A Unifying Look at Data Structures," *Communications of the ACM* **23**(4), April 1980, 229–239.

**Imprimé en France**

**par**

**l'Institut National de Recherche en Informatique et en Automatique**