Page usage in quadtree indexes

M. Hoshi, Philippe Flajolet

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° 1434

Programme 2
Calcul Symbolique, Programmation
et Génie logiciel

PAGE USAGE IN QUADTREE INDEXES

Mamoru HOSHI
Philippe FLAJOLET

Mai 1991

Page Usage in Quadtree Indexes

Mamoru Hoshi Philippe Flajolet

Abstract. *This paper provides a characterization of the storage needs of quadtrees when used as indexes to access large volumes of 2-dimensional data. It is shown that the page occupancy for data in random order approaches 33%. A precise mathematical analysis that involves a modicum of hypergeometric functions and dilogarithms, together with some computer algebra is presented.*

A brief survey of the analysis of storage usage in tree structures is included. The 33% ratio for quadtrees is to be compared to the figures for binary search trees (50%), tries (69%), and quadtries (46%).

De l'occupation des pages dans les arbres Quad utilisés comme index

Résumé. Cet article propose une caractérisation des besoins en mémoire d'arbres Quad lorsque ceux-ci sont utilisés en tant qu'index afin d'accéder de grandes quantités d'informations en 2-dimensions. L'on montre que le taux d'occupation des pages, pour un modéle d'ordre aléatoire des données, avoisine 33%. En est présentée une analyse mathématique précise qui met en jeu quelques fonctions spéciales, hypergeométriques et dilogarithmes, ainsi que l'utilisation du calcul formel.

L'artice conclut par un bref survol de la consommation mémoire des structures d'arbres. Le taux de 33% pour les arbres Quad est à comparer aux chiffres des arbres binaires de recherche (50%), des treilles digitales (69%), ou des treilles Quad (46%).

Page Usage in Quadtree Indexes

Mamoru Hoshi*
Faculty of Engineering,
Chiba University,
I-33 Yayoi-Cho, Chiba
Japan 260

Philippe Flajolet†
Algorithms Project,
INRIA, Rocquencourt,
F–78153 Le Chesnay
France

May 15, 1991

Abstract. This paper provides a characterization of the storage needs of quadtrees when used as indexes to access large volumes of 2–dimensional data. It is shown that the page occupancy for data in random order approaches 33%. A precise mathematical analysis that involves a modicum of hypergeometric functions and dilogarithms, together with some computer algebra is presented.

A brief survey of the analysis of storage usage in tree structures is included. The 33% ratio for quadtrees is to be compared to the figures for binary search trees (50%), tries (69%), and quadtries (46%).

1 Introduction

The quadtree structure is a fundamental hierarchical representation of point data in higher dimensional spaces. It was invented by Finkel and Bentley in 1974 [6], and it is a natural generalization of binary search trees to multidimensional data. Under one form or the other, it has surfaced in many different fields, like data bases, geographical data processing, graphics and image processing. A comprehensive treatment of this area of algorithmic design is to be found in Samet's book [21].

*The research of this author was done while visiting INRIA, Rocquencourt, France under support from the Ministry of Education of Japanese Government.

†Work of this author was supported in part by the Basic Research Action of the E.C. under contract No. 3075 (Project ALCOM).

We discuss here the (point) quadtrees, for data in 2–dimensional space. More precisely, we concentrate on quadtrees that depend on an integer parameter $b \geq 0$ representing a *page capacity*, sometimes also called a *bucket* capacity; small subfiles (*i.e.*, with size $\leq b$) are represented sequentially into a page instead of being split recursively.

The paged quadtrees that we consider will thus naturally occur if one needs to maintain large collections of data on external storage using the quadtree principle. They can also be useful even as direct (in core) data structures since they build a hierarchical cell decomposition: If b is large enough, nearest neighbours of a point are very likely to be found in the same cell (page); in this way nearest neighbour queries can be answered by a simple local search which is fairly efficient and adaptive.

Our major results characterize the expected storage occupancy of quadtrees. For data in random order, we establish that the filling ratio of pages is approximately 33%, in the sense that the number of pages necessary to store a file of n points with b the page capacity is about

$$\frac{3n}{b}.$$

Our precise results are the following.

Theorem 1 *Given a page capacity $b \geq 1$, there exists a constant γ_b such that the expected number of pages for a paged quadtree with page capacity b built on n random points satisfies*

$$P_n^{[b]} = \gamma_b \cdot n + O(\log n), \tag{1}$$

where γ_b is

$$\frac{1}{3}\gamma_b = 3\int_0^1 \frac{(1-t)^3}{t(1+2t)^2}\,dt \int_0^t \frac{(1+2v)}{(1-v)^2} E_b(v)\,dv \tag{2}$$

with

$$E_b(z) = z^b \frac{1 + b(1-z) + b(b+1)(1-z)^2}{(1-z)^2}.$$

From this theorem, we can determine the values of the constant γ_b. (We also give the values of $b\gamma_b$.)

b	γ_b	$b\gamma_b$
0	3	
1	1.564747	1.56475
2	1.041362	2.08272
3	0.776966	2.33090
4	0.618679	2.47472
5	0.513623	2.56812
10	0.277208	2.77209
15	0.189691	2.84537
20	0.144151	2.88302
25	0.116237	2.90593
30	0.0973780	2.92134
35	0.0837832	2.93241
40	0.0735188	2.94075
45	0.0654947	2.94726
50	0.0590496	2.95248

It may be of interest to note that the table above does *not* result from straight numerical integration, which would be conducive to various numerical difficulties. Its derivation was first based instead on symbolic integration performed by the Maple system [3]: For values 1-10 and 15(5)50, the computation took a little over 600 seconds of CPU time (on a Sun3 machine). For instance, we have for $b = 50$, the verbatim form of γ_b,

```
3159614683170552814765839048751265660686349                2
---------------------------------------------- - 390150 Pi
  8205456738260767651760056073099788 80
```

The first values are given below.

b	γ_b
0	3
1	$120 - 12\pi^2$
2	$534 - 54\pi^2$
3	$1422 - 144\pi^2$
4	$\frac{5923}{2} - 300\pi^2$
5	$\frac{53301}{10} - 540\pi^2$
10	$\frac{252794897}{7056} - 3630\pi^2$

All these numerical data suggest definite patterns: γ_b is a rational function of π, the coefficient of π^2 has a simple form, and $\gamma_b \approx 3/b$ for large b. In effect, we have:

Theorem 2 *(i). The coefficient γ_b is a linear function of π^2,*

$$\frac{1}{3}\gamma_b = 6b^2 + 9b + 1 - 6b(b+1)^2[\frac{\pi^2}{6} - \sum_{j=1}^{b} \frac{1}{j^2}]. \tag{3}$$

(ii). Asymptotically, for large b, we have[1]

$$\frac{1}{3}\gamma_b = \frac{1}{b} - \frac{4}{5\,b^2} + \frac{2}{5\,b^3} + \frac{2}{35\,b^4} - \frac{2}{7\,b^5} + \frac{2}{35\,b^6} + \frac{2}{5\,b^7} - \frac{14}{55\,b^8} + O(\frac{1}{b^9}).$$

On the practical side, we would like to comment on this 33% page filling ratio. Often, for a data structure, a relatively low filling ratio can be obviated by a suitable allocation policy. Assume for instance, that we choose to implement a paged quadtree structure which we design with a parameter $b = 60$; the pages created are called "logical" pages. If we allocate *physical* pages of capacity $\beta = 20$, the quadtree structure built with logical pages with parameter $b = 60$ will have each of its logical pages spread over 1, 2, or 3 physical pages. Our analysis (see Section 4 and Theorem 5) enables us to quantify precisely what happens: In that situation, the number of disk accesses increases slightly and it is on average 1.421145; in counterpart, the (physical) filling ratio improves appreciably and becomes close to 0.67273. In summary we more than double the occupancy rate at the expense of an increase of less than 50% of the access time.

Thus, the analysis techniques developed here are of some level of generality, since they apply to a fairly general class of additive cost measures on quadtrees: Theorem 4 discusses statistics on arbitrary node types in quadtrees; as a particular application, we are able to characterize the expected number of pages containing k elements ($0 \leq k \leq b$), and thus attain a precise evaluation of the page occupancy profile in paged quadtrees.

The evaluation of filling ratios is useful in order to assess and possibly optimize various allocation strategies. In this spirit, the paper concludes with a brief survey of analytical results available for tree indexes of various sorts.

To a large extent our Theorem 2 owes its existence to the integration capabilities of the Maple system for computer algebra [3] which first revealed the unsuspected occurrence of closed form expressions involving dilogarithms and made it possible to carry out easily rather intensive computations.

2 Paged Quadtrees

Our data model assumes data in random order. Without loss of generality, we take them independently and uniformly distributed over the unit square $\mathcal{Q} = [0,1] \times [0,1]$. Given a sequence $S = (S_1, S_2, \ldots, S_n)$ of points, $S \in \mathcal{Q}^n$, we form a tree, called a *b–quadtree*, by the following rules:

- If $|S| \leq b$, then the tree consists of a single external (page) node that contains S itself.
- If $|S| > b$, then the first element S_1 of S partitions the other elements $(S_2, \ldots, S_n)$ into four subsequences, based on the four quadrants (*North–West, North–East*, etc.) determined by S_1, namely $S_{NW}, S_{NE}, S_{SW}, S_{SE}$. The tree associated to S is composed

[1]The absolute errors provided by the approximate formula obtained by dropping the $O(.)$ error terms are of order respectively $10^{-3}, 10^{-5}, 10^{-8}$ for $b = 2, 4, 8$.

of a root which contains S_1 and of the four subtrees formed recursively from the four subsequences $S_{NW}, S_{NE}, S_{SW}, S_{SE}$.

The standard quadtree of Finkel and Bentley appears when $b = 0$, and one singles out the external empty nodes. A b–quadtree can be alternatively viewed as a standard quadtree in which maximal subtrees of size $\leq b$ are grouped into individual pages. With this view, the number of pages or the number of internal nodes of a paged b–quadtree are simple parameters of the underlying standard quadtree. Our paper is in fact a paper on cost measures on standard quadtrees applied to paging.

Notations. Given a sequence of numbers $\{f_n\}_{n\geq 0}$, its *generating function* (GF) is

$$f(z) = \sum_{n\geq 0} f_n z^n.$$

We also use $[z^n]\, f(z)$ in order to represent the coefficient of z^n in $f(z)$, that is $[z^n]f(z) = f_n$.

Additive functions over quadtrees. We consider here a general additive function over *standard* quadtrees

$$\begin{cases} f[t] &= e_{|t|} + \sum_{j=1}^{4} f[t_j] \\ f[\emptyset] &= e_0, \end{cases} \tag{4}$$

with t_1, t_2, t_3, t_4 the root subtrees of t; there e_n is a sequence of numbers, called the "tolls". Thus $f[t]$ represents the total cost associated to a tree, when there is a toll (depending on subtree sizes) at each node in the tree.

For instance, if the toll is $e_n \equiv 1$, then $f[t]$ is the total number of nodes in the tree; if $e_n = n$, we get the path length of the tree. Given the paging parameter b, the number of internal nodes in the associated b–quadtree, corresponds clearly to the toll function

$$e_n = 1 \text{ if } n > b; \qquad e_n = 0 \text{ if } 0 \leq n \leq b. \tag{5}$$

In this case, the number of external nodes (*i.e.*, pages) is $3f[t] + 1$, because of the general conservation law on quaternary trees.

In the sequel, we keep $f[t]$ in order to denote a generic tree cost, we reserve $I[t]$ and $P[t] = 3I[t] + 1$ for the number of internal and external nodes, when the parameter b has been fixed.

If $f[.]$ is a cost, we let f_n be its expectation, when taken over all randomly built quadtrees over n data items. The generating functions of the sequences $\{e_n\}$ and $\{f_n\}$ are thus

$$e(z) = \sum_{n\geq 0} e_n z^n \qquad f(z) = \sum_{n\geq 0} f_n z^n.$$

Lemma 1 *Let $\{e_n\}$ be a toll sequence with $e_0 = 0$; let f_n be the expectation of the corresponding cost as defined by Eq. (4). Then the associated GF's $e(z)$ and $f(z)$ are related by*

$$f(z) = \frac{(1+2z)}{(1-z)^2} \int_0^z \frac{(1-t)^3}{t(1+2t)^2}\, dt \int_0^t \frac{(1+2v)}{(1-v)^2} E(v)\, dv \tag{6}$$

where $E(z)$ is the modified cost generating function,

$$E(z) = \frac{d}{dz} z(1-z) \frac{d}{dz} e(z).$$

PROOF. Let $\pi_{n,k}$ denote the probability that a quadtree of size n has its first (*e.g.*, *NW*) subtree of size k. We have [4, 8, 16]

$$\pi_{n,k} = \frac{1}{n} \sum_{\ell=k}^{n-1} \frac{1}{\ell+1}.$$

An informal interpretation is that each of the n possibilities, $\{0, 1, \ldots, n-1\}$, for the number of elements going to West is equally likely and has probability $1/n$; if ℓ elements are located West of the root, then each value $K \in [0..\ell]$ of the number of elements residing North-West is equally likely and has probability $1/(\ell+1)$. (We refer the reader to the cited publications for more convincing arguments!)

With this form of the $\pi_{n,k}$, the standard recurrence for costs is

$$f_n = e_n + 4 \sum_{k=0}^{n-1} \pi_{n,k} f_k, \tag{7}$$

where we have taken advantage of obvious symmetries.

Thus, if we go to the realm of generating functions, we find the integral equation that corresponds to (7),

$$f(z) = e(z) + 4 \int_0^z \frac{dt}{t(1-t)} \int_0^t f(u) \frac{du}{1-u}. \tag{8}$$

By differentiations, we get the equivalent differential equation,

$$z(1-z) \frac{d^2}{dz^2} f(z) + (1-2z) \frac{d}{dz} f(z) - \frac{4}{1-z} f(z) = E(z), \tag{9}$$

where

$$E(z) = \frac{d}{dz} z(1-z) \frac{d}{dz} e(z).$$

First, one looks at the homogeneous equation defined by setting $E(z) = 0$ inside (9).

One method[2] consists in solving this equation by reducing it to a degenerate hypergeometric equation, as was done for similar problems in [8]: We look for an approximate solution of the form $(1-z)^\alpha$, find the indicial equation $\alpha^2 - 4 = 0$ so that $\alpha = \pm 2$, try a solution

[2]It is interesting to note that the equation is now in principle solvable by general purpose algorithms that determine rational solutions to linear ODE's, see *e.g.* [2]. Some amount of human interaction is however still needed since we impose additional analyticity requirements around 0. Also, the general reduction of a quadtree analysis to hypergeometric equations is an especially effective and general tool [8], so that we have decided to reduced ourselves to this treatment instead of directly invoking a *deus ex machina* formula, $\phi(z) = (1+2z)/(1-z)^2$.

of the form $\hat{f}(z)(1-z)^{-2}$, and observe that $\hat{f}$ satisfies a standard hypergeometric equation [25],

$$z(1-z)\frac{d^2}{dz^2}\hat{f}(z) + (1+2z)\frac{d}{dz}\hat{f}(z) - 2\hat{f}(z) = 0,$$

which admits the special (hypergeometric!) solution $\hat{f}(z) = (1+2z)$.

The whole process thus provides us with the particular solution

$$\phi(z) = \frac{\hat{f}(z)}{(1-z)^2} = \frac{1+2z}{(1-z)^2} \qquad \text{when} \qquad E(z) = 0,$$

another independent solution being discarded as it has a logarithmic singularity at 0.

Returning then to the inhomogeneous equation, we proceed by the variation of constant method. We seek a solution of the form $\lambda(z)\cdot\phi(z) = \lambda(z)(1+2z)/(1-z)^2$. By construction, $\lambda'(z)$ satisfies an ODE of order 1, hence, we recover the solution to the original equation by two quadratures, the result being as stated above. ∎

Paging. If we specialize to the case of the number of pages in a b-quadtree, we get:

Lemma 2 *The generating function for the expected number of pages in a b-quadtree is*

$$P(z) = \frac{1}{1-z} + \frac{3(1+2z)}{(1-z)^2}\int_0^z \frac{(1-t)^3}{t(1+2t)^2}\,dt \int_0^t \frac{(1+2v)}{(1-v)^2}E(v)\,dv, \tag{10}$$

with

$$E(z) \equiv E_b(z) = z^b \frac{1+b(1-z)+b(b+1)(1-z)^2}{(1-z)^2}.$$

PROOF. This is a simple application of the previous lemma. The tolls for the number of internal nodes in a b-quadtree are the e_n given above (5), with GF equal to

$$e(z) = \frac{z^{b+1}}{1-z} \qquad \text{and} \qquad E(z) = \frac{d}{dz}z(1-z)\frac{d}{dz}\frac{z^{b+1}}{1-z}.$$

We derive in this way $I(z)$ by Lemma 1. By the conservation law of quaternary trees, we finally have $P(z) = 3I(z) + 1/(1-z)$. ∎

This expression would in principle enable us to express in "closed form" the average number of pages (see [4, 16] for related computations done independently via a recurrence approach). We prefer however a direct route to asymptotics based on the usual method of *singularity analysis* [9].

> **Singularity Analysis.** The general principle is that the asymptotic behaviour of coefficients $[z^n]P(z)$ can be determined from the asymptotic form of function $P(z)$ around its dominant singularities. The conditions are based on analytic continuation. They make it possible to transfer on a term-by-term basis from asymptotic elements of $P(z)$ to matching asymptotic elements of $[z^n]P(z)$.

Here, from either the differential equation and general theorems [24], or more explicitly from the integral representations, we see that $P(z)$ has a unique isolated logarithmic singularity at $z = 1$. Thus $P(z)$ is analytically continuable outside of its circle of convergence, say in $|z| < 2, |\mathrm{Arg}(z-1)| > \pi/4$. Also, from the integral representation, there results that, in this region,

$$P(z) = \frac{\gamma_b}{(1-z)^2} + O((1-z)^{-1}\log(1-z)^{-1}) \qquad (z \to 1).$$

By the techniques of singularity analysis, this local expansion together with the analytic continuation of $P(z)$ outside its circle of convergence are enough to make legal the term–by–term transfer to coefficients, namely

$$P_n = \gamma_b \cdot n + O(\log n).$$

This therefore completes the proof of Theorem 1. ■

Leaves in quadtrees. In order to shed some light on the internals of the computation, we examine the determination of the expected number of *leaves* in a randomly grown quadtree. In that case, we have $b = 1$, and look at internal nodes. With our earlier notations, the corresponding GF is $I(z)$; the expected number of leaves is then $n - [z^n]I(z)$.

The interest of the computations that follow is to introduce a special function, namely the *dilogarithm*.

For $b = 1$, the function $E(z)$ is equal to $-1 + 2z + (1-z)^{-2}$. The inner integral $\int_0^t \ldots dv$ in (10) is then found to be

$$-\frac{t(t-2)(4t^2-7t+4)}{(1-t)^3} + 8\log(1-t).$$

Multiplying by $(1-t)^3 t^{-1}(1+2t)^{-2}$, and integrating, we find a sum of two terms, one corresponding to the rational part, the other to the logarithm. The part corresponding to the rational term is a standard elementary function.

Recall the definition of the dilogarithm as

$$\mathrm{Li}_2(z) = \int_0^z \log(1-t)^{-1}\frac{dt}{t} = \sum_{k=1}^{\infty}\frac{z^k}{k^2}. \tag{11}$$

(We refer the reader to Lewin's classic treatease for a full exposition of the theory of the dilogarithm [18] or to Berndt's review of its main properties in [1, Chap. 9].) A dilogarithm arises from integration of the logarithmic term, $8\log(1-t)$, multiplied by the element $1/t$ that comes from the partial fraction decomposition

$$\frac{(1-t)^3}{t(1+2t)^2} = \frac{1}{t} - \frac{1}{4} - \frac{27}{4(1+2t)^2}.$$

All computations done, we get

Corollary 3 *The generating function for the number of non-leave nodes in a randomly grown quadtree ($b = 1$) is*

$$I(z) = \frac{z(28 + 13z - z^2)}{(1-z)^2} + \frac{20 + 4z}{1-z}\log(1-z) - 8\frac{1+2z}{(1-z)^2}\mathrm{Li}_2(z), \tag{12}$$

with $\mathrm{Li}_2(u)$ *the dilogarithm function. Thus,*

$$[z^n]I(z) = (40 - 4\pi^2)n + 13 - \frac{4}{3}\pi^2 + + \frac{4}{3n^3} - \frac{4}{5n^4} - \frac{4}{15n^5} + \frac{4}{7n^6} + \frac{4}{21n^7} + O(\frac{1}{n^8}).$$

In particular, the proportion of leaves in a random quadtree of size n is asymptotic to $4\pi^2 - 39 = 0.47841762$.

PROOF. (Sketch) Here we obtain directly the asymptotic form $I_n \sim \gamma_1^* n$, with $\gamma_1^* = \lim_{z\to 1}(1-z)^2 I(z) = 40 - 4\pi^2$. (We have also $\gamma_1^* = \gamma_1/3$ in terms of our standard notations.) The result for leaves follows by complementation to n of the number of non-leaves. ■

An entirely similar process applies to the problem of estimating the number of pages for an arbitrary b. The occurrence of the dilogarithm which satisfies

$$\mathrm{Li}_2(1) = \int_0^1 \log(1-t)^{-1}\frac{dt}{t} = \sum_{k=1}^{\infty}\frac{1}{k^2} \equiv \frac{\pi^2}{6},$$

"explains" the presence of π^2 in the explicit forms of γ_b given in the introduction. We shall see that such a treatment can be extended to arbitrary node types.

From the exact form of $I(z)$, we also observe that the coefficient $[z^n]I(z)$ is expressible in terms of the harmonic number $\zeta_n(1)$ and the generalized harmonic number $\zeta_n(2)$, where

$$\zeta_n(s) = \sum_{k=1}^{n}\frac{1}{k^s}.$$

Such expressions were obtained by Laforest *et al.* [15, 16] using a direct theory of quadtree recurrences from [4] which constitutes an alternative to our Lemma 1.

We are going to elicit the finer structure of γ_b as a function of b in the next section.

3 The occupancy constants γ_b

Our approach now consists in computing the generating function of the numbers γ_b. The following lemma provides a more direct access to the numbers γ_b that avoids integration, and also proves that γ_b has a rational expression in terms of π^2. Analysing the singularity of the GF of the γ_b further provides detailed asymptotic informations on these coefficients.

Lemma 3 *The generating function $\gamma(u)$ of the numbers γ_b defined by $\gamma(u) = \sum_{b=0}^{\infty} \gamma_b u^b$ is given by*

$$\gamma(u) = \frac{3}{(1-u)^4} \cdot \Big[(-4u - 2u^2)\pi^2 + (1 + 30u - 27u^2 - 4u^3)$$

$$+(-6 - 24u + 30u^2)\log(1-u) + (24u + 12u^2)\mathrm{Li}_2(u)\Big].$$

PROOF. Define the basic integrals

$$J_\alpha(u) = \int_0^1 \frac{(1-t)^3}{t(1+2t)^2} dt \int_0^t \frac{(1+2v)}{(1-v)^{4-\alpha}} \frac{dv}{(1-uv)}.$$

These serve as the basis in which to express the generating function $\gamma(u)$. From the summations

$$\sum_m u^m v^m = \frac{1}{1-uv}, \quad \sum_m m \cdot u^m v^m = u\frac{d}{du}\frac{1}{1-uv}, \quad \sum_m m(m-1) \cdot u^m v^m = u^2 \frac{d^2}{du^2}\frac{1}{1-uv},$$

and the integral representation of γ_b, we find that

$$\frac{1}{3}\gamma(u) = J_0(u) + u\frac{d}{du}J_1(u) + u^2\frac{d^2}{du^2}J_2(u) + 2u\frac{d}{du}J_2(u).$$

Our problem is thus reduced to computing the quantities J_0, J_1, J_2.

In principle, the problem resembles the computation in our earlier section, see for instance the particular case of counting leaves. It is however complicated by the extra factor $(1-uv)^{-1}$ that introduces an additional singularity in the computations.

Preliminary investigations performed with the Maple system first revealed the possibility of an explicit solution that involves dilogarithms. Once this has been recognized, it is possible to carry out the double integration. Minor computer algebra difficulties arise from several sources: certain normal forms provided by integration routines sometimes introduce transformations of the form $\log(1-t) \mapsto \log(t-1) + \log(-1)$; the solutions, though representing generating functions, may have apparent singularities at 0 that need to be eliminated; finally, some of the expressions obtained involve the dilogarithm under a form that is singular at 0.

We dispense ourselves from giving here all the explicit forms of the J_α and the partial integrals involved. Once found by whatever means, they are all that is needed in order to reconstruct a complete proof of the expression given for $\gamma(u)$, since the correctness of integrals can always be established by differentiation. We only indicate in the appendix a sequence of steps needed to obtain $J_0(u)$ using the Maple system.

In passing, the solution there is expressed in terms of Maple's version of the dilogarithm function

$$\texttt{dilog}(u) = \mathrm{Li}_2(1-u) = \sum_{k=1}^{\infty} \frac{(1-u)^k}{k^2}.$$

The reduction to a standard dilogarithm, evaluated near 0, is achieved via the well known transformation formula (whose proof is a single integration by parts):

$$\mathrm{Li}_2(1-z)+\mathrm{Li}_2(z)=\frac{\pi^2}{6}-\log z\log(1-z). \tag{13}$$

From this, the proof of the lemma follows. ■

From Lemma 3, explicit forms of the γ_b are derived. The principle is to express the GF $\gamma(u)$ in the basis of functions

$$h_{1,j}(u)=\bar{\theta}^j\frac{\log(1-u)^{-1}}{(1-u)} \qquad \text{and} \qquad h_{2,j}(u)=\bar{\theta}^j\frac{\mathrm{Li}_2(u)}{(1-u)},$$

where $\bar{\theta}$ represents the differential operator $\bar{\theta}\{f(u)\}\equiv\frac{d}{du}\{uf(u)\}$, the coefficients of these functions involving generalized harmonic numbers, since

$$\frac{1}{1-u}\mathrm{Li}_2(u)=\sum_{n=0}^{\infty}\zeta_n(2)u^n.$$

We find

$$\frac{1}{3}\gamma(u)=6[\bar{\theta}^3-\bar{\theta}^2]\{\frac{1}{1-u}\mathrm{Li}_2(u)\}+\frac{1+13u-2u^2}{(1-u)^3}-2\pi^2\frac{u(2+u)}{(1-u)^4}. \tag{14}$$

It is an easy matter to expand $\gamma(u)$ from this form. This completes the proof of Part (i) of Theorem 2.

The asymptotic form of γ_b next results from singularity analysis. There is a full asymptotic expansion of $\gamma(u)$ around $u=1$. The term $\mathrm{Li}_2(u)$ is expanded using the basic functional equation (13). In this way, we find

$$\frac{1}{3}\gamma(u)=[\mathrm{Li}_1(u)+\frac{1}{12}]+(1-u)[\frac{4}{5}\mathrm{Li}_1(u)+\frac{17}{75}]+(1-u)^2[\frac{3}{5}\mathrm{Li}_1(u)+\frac{17}{100}]+\cdots \tag{15}$$

where $\mathrm{Li}_1(u)=\log(1-u)^{-1}$. Using the identity

$$[u^m](1-u)^k\mathrm{Li}_1(u)=\frac{(-1)^k k!}{m(m-1)(m-2)\cdots(m-k)},$$

we map the singular expansion (15) into a matching expansion for $\gamma_m=[u^m]\gamma(u)$, the conditions of analytic continuation being clearly satisfied here. In this way, we get

$$\frac{1}{3}\gamma_m=\frac{1}{m}-\frac{4}{5}\cdot\frac{1!}{m(m-1)}+\frac{3}{5}\cdot\frac{2!}{m(m-1)(m-2)}-\cdots,$$

which can be normalized into a standard expansion into descending powers of $1/m$. This completes the proof of Part (ii) of Theorem 2. ■

4 Node types

The same methods make it possible to analyze the number of occurrences of nodes of arbitrary composition in quadtrees. Assume we look for the expected number of nodes ν in a random tree of size n such that the subtree rooted at ν has a fixed shape ω. This corresponds to a toll sequence $\hat{e}_n$ such that $\hat{e}_n = 0$ for all values of $n \neq |\omega|$. For $p = |\omega|$, $\hat{e}_p$ is a rational number equal to the probability that the tree shape ω occurs as randomly built quadtree on p elements. That probability is computable inductively over subtrees using the form of splitting probabilities [8]

$$\pi_{n_1,n_2,n_3,n_4} = \frac{1}{n \cdot n!} \frac{(n_1+n_2)!\,(n_3+n_4)!\,(n_1+n_3)!\,(n_2+n_4)!}{n_1!\,n_2!\,n_3!\,n_4!},$$

which represents the probability that the (NW, NE, SW, SE) root subtrees have respective sizes n_1, n_2, n_3, n_4. If $\omega = \langle r; t_1, t_2, t_3, t_4 \rangle$ is a tree with root r and the t_j's as root subtrees, we have

$$\epsilon_\omega = \pi_{|t_1|,|t_2|,|t_3|,|t_4|} \cdot \epsilon_{\omega_1} \epsilon_{\omega_2} \epsilon_{\omega_3} \epsilon_{\omega_4},$$

together with the initial conditions $\epsilon_\omega = 1$ if $|\omega| \leq 1$.

Thus, we find the toll generating function $\hat{e}(z) = \epsilon_\omega z^p$, with $p = |\omega|$, where ϵ_ω is an easily computable rational number. If we compare this to the toll GF considered earlier in connection with paging, $e_b(z) = z^{b+1}/(1-z)$, we see that

$$\hat{e}(z) = \epsilon_\omega [e_{|\omega|-1}(z) - e_{|\omega|}(z)].$$

By linearity of the cost transform (Lemma 1), we get:

Theorem 4 *Consider an arbitrary node type defined by a tree shape ω. The expected number of nodes of type ω admits the asymptotic form*

$$n \cdot \frac{\epsilon_\omega}{3} [\gamma_{|\omega|-1} - \gamma_{|\omega|}],$$

where $\epsilon_\omega \in \mathbf{Q}$ is the probability of tree shape ω amongst all quadtrees of size $|\omega|$.

The coefficients are therefore $\mathbf{Q}$-linear combinations of 1 and π^2.

This generalizes results of Laforest *et al.* [16, 15] who studied nodes having a single child. (Full asymptotic expansions for the number of nodes of a given type could also be obtained in the style of Corollary 3.) As a check, we can also retrieve the expected number of leaves, corresponding to $|\omega| = 1$, which leads to the asymptotic form $\frac{n}{3}(\gamma_0 - \gamma_1)$.

The γ_b thus appear as fundamental constants in the analysis of quadtrees. From them, one can determine the *profile* of page occupancy.

Theorem 5 *In a paged b-quadtree, the expected number of pages containing k elements, $0 \leq k \leq b$, is of the asymptotic form $\gamma_{b,k} \cdot n$, with*

$$\gamma_{b,k} = \frac{\gamma_b}{b+1} + B[H_{b+1} - 1 - H_k], \qquad B = \frac{2}{3} \cdot \frac{3b\gamma_b + 2\gamma_b - 6}{b(b+1)},$$

where $H_n \equiv \zeta_n(1) = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$ is the standard harmonic number.

PROOF. As an application of Theorem 4, we first count the expected number of pages that satisfy the conditions: (i) they are leftmost child; (ii) they contain k elements; (iii) their father is the root of a subtree with m elements for some fixed $m > b$. Using the form of the splitting probabilities $\pi_{m,k} = (H_m - H_k)/m$, we find that the asymptotic proportion of such pages is

$$\frac{H_m - H_k}{3m}[\gamma_{m-1} - \gamma_m].$$

The constant $\gamma_{b,k}$ is obtained by multiplying by 4 (to take care of all four child nodes) and summing over all values of m from $b+1$ to ∞. In this way, we see that

$$\gamma_{b,k} = A - BH_k, \qquad A = \sum_{m=b+1}^{\infty} \frac{4H_m}{3m}[\gamma_{m-1} - \gamma_m], \quad B = \sum_{m=b+1}^{\infty} \frac{4}{3m}[\gamma_{m-1} - \gamma_m]. \tag{16}$$

The constants A, B could probably be found by direct summation. It is however simpler, once their existence has been recognized, to identify them by means of conservation laws for nodes. We have

$$\sum_{k=0}^{b} \gamma_{b,k} = \gamma_b \qquad \text{and} \qquad \sum_{k=0}^{b} k \cdot \gamma_{b,k} = 1 - \frac{\gamma_b}{3}. \tag{17}$$

The first relation expresses that a page contains a certain number k of elements for some $k \in [0..b]$; the second relation consists in estimating the proportion of elements contained in pages either as non-internal elements (whose proportion is $1 - \gamma_b/3$) or based on the size of the page that contains them.

We use the easy relations

$$\sum_{k \le b} H_k = (b+1)(H_{b+1} - 1), \qquad \sum_{k \le b} kH_k = \frac{1}{2}b(b+1)H_{b+1} - \frac{1}{4}b(b+1),$$

and then solve for A and B the system (17). In this way, we obtain the values of A, B and the statement of the theorem follows. ■

For instance for $b = 10$, we find the following proportions

$$\begin{aligned} &\gamma_{10,0} = 0.06034, \ \gamma_{10,1} = 0.04294, \ \gamma_{10,2} = 0.03424, \ \gamma_{10,3} = 0.02844, \\ &\gamma_{10,4} = 0.02409, \ \gamma_{10,5} = 0.02061, \ \gamma_{10,6} = 0.01771, \ \gamma_{10,7} = 0.01523, \\ &\gamma_{10,8} = 0.01305, \ \gamma_{10,9} = 0.01112, \ \gamma_{10,10} = 0.00938. \end{aligned}$$

All these constants have again exact forms that are expressible as functions of π^2. It is from them that we can analyze arbitrary page allocation strategies (see, e.g., the example given in the introduction with $b = 60$ and $\beta = 20$).

5 Conclusions

We conclude this paper with a brief overview of some major algorithms for maintaining dynamic tree structures in a paging environment. There are two major categories since

structures are built either based on order properties of the data—the *comparison based* data structures—or on *digital* properties. Some of the trees are of fixed degree (2 or 4 depending on the dimension of the data space: binary search trees, tries, quadtrees, etc); others have a branching degree that varies with b (e.g., for B-trees it varies between $b/2$ and b; for m-ary search trees, it is equal to m with $m = b + 1$, etc.). We refer to either Sedgewick's book [22] or to Gonnet's encyclopedia [12] as general sources on the algorithmic aspects. Average case analysis techniques are reviewed in [23].

Each analysis of storage occupancy normally poses an interesting mathematical problem. In this quick review, we also mention the major mathematical techniques at stake.

Comparison–based structures. Binary search trees [14, Sec. 6.2.2] are the simplest structures to analyse. We consider the strategy already discussed for quadtrees whereby a maximal subtree of size $\leq b$ is stored into a single page. It is then found that the expected number of pages is asymptotic to $2n/(b+2)$. In other words, storage occupancy is near 50%. The generating function equations are simpler in this case. The main equation is of the form

$$f(z) = e(z) + 2\int_0^z f(t)\frac{dt}{1-t}.$$

This reduces to a *differential equation* of order 1 that can be solved by quadratures. Many parameters can be analyzed in this way by varying the "toll" GF. The model is the same as the one underlying Quicksort, see Knuth's book [14, p. 121] and Hennequin's thesis [13]. In particular, we find that the number of pages containing r elements is $\sim 2n/((b+1)(b+2))$ for $r \in [0..b]$: In other words, pages with filling type $0/b, 1/b, \ldots, b/b$ are all equally frequent.

The storage occupancy of search trees whose degree is $m = b + 1$ (a node contains b keys and $b + 1$ pointers) is investigated extensively by Mahmoud and Pittel [19]. The cost generating function satisfies a linear differential equation of order $b - 1$, namely

$$\frac{d^b}{dz^b}f(z) = e(z) + \frac{(b+1)!}{(1-z)^b}f(z).$$

The analysis is made possible because there is a regular singularity at $z = 1$. It is found (see also [14, Ex. 6.2.4.10]) that the number of nodes in the tree is on average

$$\sim \frac{n}{2(H_{b+1}-1)} \qquad \text{where } H_m = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{m},$$

a harmonic number: Storage utilization tends to 0 as b get large! In fact Mahmoud and Pittel obtain asymptotic distribution results, a rather remarkable fact, since this requires analyzing a non-linear difference differential equation of high order.

The efficiency of m-ary search trees ($m = b + 1$) gets quite low as b becomes large. Balancing is however a good solution with guaranteed worst case performance (at worst 50%). Yao has shown that for balanced B-trees of large order the storage occupancy rate approaches $\log 2$, and the number of nodes is approximately $\approx \frac{n}{b \log 2}$. Yao's paper [26] is well

	Comparison based	Digital
1-dim	Binary search tree, [0.5] $2n/(b+2)$	Binary digital trie, [0.69] $n/(b\log 2)$
	m–ary search tree, [0.0] $(m=b+1):\quad n/(2(H_m-1))$	Paged b–digital tree, [0.69] $\approx n/(b\log 2)$
	Balanced B-tree, [0.69] $n/(b\log 2)$	
		Extendible Hash directory, [0.0] $\approx 4b^{-1}n^{1+1/b}$
2-dim	Quadtree, [0.33] $\approx 3n/b$	Quadtrie, [0.46] $3n/(2b\log 2)$
		Grid file directory, [0.0] $\approx n^{1+1/(2b+1)}$

Figure. A summary of some major paging strategies for trees and their expected performance in asymptotic form. There n is the file size, and b represents the page capacity in terms of records that a page can contain. The number in brackets, $[\rho]$, represents a numerical aproximation of the filling ratio ρ such that the expected storage occupancy varies like $n/(b\rho)$.

known as the source of so–called fringe analyses that are based on Markovian approximations and matrix analysis.

Our results regarding quadtrees are based on an integral transform (Lemma 1) that permits to resolve algebraically a class of cost functions on quadtrees; they further rely on singularity analysis and on special functions (the hypergeometric equation, the dilogarithm). Quite clearly, the approach taken here is general and applies to almost any conceivable additive parameters on quadtrees.

Digital methods. Digital methods use a separation principle based on bits of records (or their hashed values). The paging of small subfiles is analyzed by Knuth using methods partly suggested by de Bruijn, see Section 6.3 of [14] and the methods of pages 131ff. there. The equations are *difference equations* of the form

$$f(z) = e(z) + 2e^{z/2}f(z/2).$$

The treatment relies on iteration and Mellin transforms. The number of pages in a trie involves some small oscillating terms, and neglecting them, it can be approximated by $\frac{n}{b\log 2}$, refer to Exercice 6.3.20 of [14], and read between the lines. The analysis is also relevant to dynamic hashing schemes [5, 17]. The same analytic principles apply to quadtries whose evaluation is isomorphic to that of m–ary tries for $m = 4$.

The digital tree structure can be extended by letting nodes contain up to b elements, but still retaining the binary branching principle. The corresponding equation becomes a difference–differential equation

$$\frac{d^b}{dz^b} f(z) = e(z) + 2e^{z/2} f(z/2).$$

Mellin transforms and singularity analysis are the main ingredients of that analysis. Apart from fluctuations, the number of pages is found [11] to be of the form $\frac{n}{b \log 2}$. Thus, the ratio of 69% strikes again here.

For completeness, we have also tabulated some of the formulæ for extendible hashing and grid files access methods. They concern the size of the directory which exhibits a non–linear growth of the form n^β, $\beta > 1$. However, the non–linearity factor is of the rough form $n^{1/b}$, so that the observed behaviour is practically linear provided small values of b are avoided. The estimates are due to Flajolet [7] and Régnier [20]. They are based on occupancy statistics, saddle point estimates and Mellin transforms.

Results in this paper indicate that, under paging conditions, trees of low degree (binary search trees and tries, quadtrees and quadtries, generalized digital trees) compare very favorably to trees with high branching degree, except when balancing can be maintained. A variety of methods from discrete mathematics have surfaced in the analysis of storage occupancy for tree data structures. The methods employed here constitute yet another illustration of the power of differential equations in conjunction with singularity analysis techniques in the area of the average case analysis of algorithms which were introduced in [10].

Acknowledgements. The authors would like to acknowledge the constant help and support of the Maple system that might well have been a coauthor of the paper. By suggesting the existence of closed form in terms of dilogarithms, Maple made it possible for us to ultimately derive Theorems 2, 4, and 5 in a clean form.

Mamoru Hoshi would like to express his gratitude to the members of Project ALGO at INRIA for their encouragements.

References

[1] Bruce C. Berndt. *Ramanujan's Notebooks, Part I.* Springer Verlag, 1985.

[2] Manuel Bronstein. On solutions of linear ordinary differential equations in their coefficient field. Technical Report 152, Department Informatik, ETH, January 1991.

[3] B. W. Char, K. O. Geddes, G. H. Gonnet, M. B. Monagan, and S. M. Watt. *MAPLE: Reference Manual.* University of Waterloo, 1988. 5th edition.

[4] Luc Devroye and Louise Laforest. An analysis of random d–dimensional quad trees. *SIAM Journal on Computing*, 19:821–832, 1990.

[5] R. Fagin, J. Nievergelt, N. Pippenger, and R. Strong. Extendible hashing: A fast access method for dynamic files. *A.C.M. Transactions on Database Systems*, 4:315–344, 1979.

[6] R. A. Finkel and J. L. Bentley. Quad trees, a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[7] P. Flajolet. On the performance evaluation of extendible hashing and trie searching. *Acta Informatica*, 20:345–369, 1983.

[8] Philippe Flajolet, Gaston Gonnet, Claude Puech, and J. M. Robson. The analysis of multidimensional searching in quad-trees. In *Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 100–109, Philadelphia, 1991. SIAM Press.

[9] Philippe Flajolet and Andrew M. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, 3(2):216–240, 1990.

[10] Philippe Flajolet and Claude Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[11] Philippe Flajolet and Bruce Richmond. Generalized digital trees and their difference-differential equations, April 1991. 15 pages. INRIA Research Report, in press. Also submitted to *Random Structures and Algorithms.*

[12] G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures: in Pascal and C.* Addison-Wesley, second edition, 1991.

[13] Pascal Hennequin. *Analyse en moyenne d'algorithmes, tri rapide et arbres de recherche.* PhD thesis, École Polytechnique, 1991.

[14] D. E. Knuth. *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison-Wesley, 1973.

[15] Gilbert Labelle and Louise Laforest. Étude asymptotique du nombre moyen de nœuds à un enfant dans un arbre quaternaire. Technical report, LACIM, UQAM, Montreal, October 1990.

[16] Louise Laforest. Étude des arbres hyperquaternaires. Technical Report 3, LACIM, UQAM, Montreal, November 1990. (Author's PhD Thesis at McGill University).

[17] P. A. Larson. Dynamic hashing. *BIT*, 18:184–201, 1978.

[18] L. Lewin. *Polylogarithms and Associated Functions.* North-Holland, New York, 1981.

[19] H. M. Mahmoud and B. Pittel. Analysis of the space of search trees under the random insertion algorithm. *Journal of Algorithms*, 10:52–75, 1989.

[20] Mireille Régnier. Analysis of grid file algorithms. *BIT*, 25:335–357, 1985.

[21] Hanan Samet. *The Design and Analysis of Spatial Data Structures.* Addison–Wesley, 1990.

[22] Robert Sedgewick. *Algorithms.* Addison–Wesley, Reading, Mass., second edition, 1988.

[23] Jeffrey Scott Vitter and Philippe Flajolet. Analysis of algorithms and data structures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 9, pages 431–524. North Holland, 1990.

[24] W. Wasow. *Asymptotic Expansions for Ordinary Differential Equations.* John Wiley, 1965. Reprinted by Dover, 1987.

[25] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis.* Cambridge University Press, fourth edition, 1927. Reprinted 1973.

[26] A. C-C Yao. On random 2–3 trees. *Acta Informatica*, 9(2):159–170, 1978.

APPENDIX
Computation of $J_0(u)$

We present extracts from a Maple session leading to the determination of the generating function $\gamma(u)$ of the γ_b. We concentrate here on the determination of the function $J_0(u)$ defined in the text.

We need to compute a double integral. Set

$$I_0(t) = \int_0^t \frac{(1+2v)}{(1-v)^4}\frac{dv}{1-uv},$$
$$I_1(z) = \int_0^z \frac{(1-t)^3}{t(1+2t)^2} I_0(t)\,dt.$$

The Maple instructions are

```
> I0:=int((1+2*v)/(1-u*v)/(1-v)^4,v=0..t);          # result has size 738
> I1:=factor(int(I0*(1-t)^3/t/(1+2*t)^2,t=0..1));  # result has size 438
```

One must estimate $I_1(1)$ by an indirect use of limits, in order to avoid an apparent singularity. It is also necessary to select appropriate branches of the log "by hand". (At one stage,we have to do a substitution $\sqrt{-1} \mapsto 0$, whose "validity" needs to be checked independently by series expansions!)

```
> J0:=limit(I1,z=1);
# Clean this expression with ln(-1)=I*Pi ==> 0 and ln(u-1) ==> ln(1-u).
# I.e., choose appropriate branch of the logarithms.
> J0:=subs({I=0,ln(u-1)=ln(1-u)},J0);
# Correctness may be verified via series expansions.
> series(J0,u=0);
                                  2   2                   2   3      4
  1/3 + 1/12 u + (10/3 - 1/3 Pi ) u  + (89/6 - 3/2 Pi ) u  + O(u )
```

The form of J_0 that was found is literally

```
      - 1/12 (- 4 + 24 ln(1 - u) u  + 15 u + 6 ln(1 - u) u
                          2              2                2
               + 24 ln(1 - u) u  ln(u) + 24 u  dilog(u) - 54 u
                          3              3                 3
               + 12 ln(1 - u) u  ln(u) + 43 u  - 30 ln(1 - u) u
                    3              /        4
               + 12 u  dilog(u)) /  (u - 1)
                               /
```


Imprimé en France
par
. l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399