UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

A Trivial Algorithm Whose Analysis Isn't: A Continuation

Ricardo A. Baeza-Yates

Research Report
CS-86-67

December 1986

A Trivial Algorithm Whose Analysis Isn't: A Continuation

Ricardo A. Baeza-Yates †

Data Structuring Group
Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L3G1

ABSTRACT

This work analyzes insertion/deletion cycles in binary search trees with three and four elements, extending previous results of Jonassen and Knuth. We compare the symmetric and asymmetric deletion algorithms, and the results show that the symmetric algorithm works better, for trees with four elements, in accordance with many empirical measures.

1. Introduction

This work extends results of Jonassen and Knuth [1] in connection with the behavior of binary search trees (BST's for short) with **three** elements under insertion/asymmetric-deletion cycles. Our analysis yields information that refutes the hypothesis that the asymmetric algorithm at the end of the cycles produces a more balanced tree. These results are consistent with the empirical data of Eppinger [2] and Culberson [3].

In particular, we analyze symmetric algorithms and a degenerate asymmetric one in three-element BST's, obtaining the asymptotic probability of the shapes of the final tree, with a large number of insertion/deletion cycles. In four-element BST's, we make an exact analysis for the asymmetric and the symmetric algorithms using a finite number of insertion/deletion pairs. Also we show that the symmetric algorithm is asymptotically better. Part of these results are included in [4].

In this work we use the terms *random insertion, random deletion,* and *randomly built tree.* A good model in the case of insertions only is to suppose that the $N!$ possible trees built with the integers $\{1, \ldots, N\}$ are equally likely. In other words, when the i-th key is inserted, the probability of its falling in any of the i intervals defined for the precedings $i-1$ keys is the same. This type of insertion is called a random insertion, and a tree built with these insertions, a randomly built tree (or random tree, for short).

In a random deletion, we choose one of the elements in the tree, each with equal probability, and we delete it by some deletion algorithm. To delete an element in a binary search tree, there exist several algorithms. If the element does not have sons, the solution is trivial. Different ways appear when there is a son. The best known algorithms are:

(i) Asymmetric [5]: If the element to delete has a right son, it is replaced by the successor (i.e. the leftmost element of the right son). Otherwise it is replaced by the left son.

† The permanent address of the author (where this work was done) is Departamento de Ciencias de la Computación, Facultad de Ciencias Fïsicas y Matemáticas, Universidad de Chile, Casilla 2777, Santiago, CHILE.

(ii) Modified Asymmetric [6]: If the element does not have a left son, it is replaced by its right son. Otherwise it is done as in (i).

(iii) Symmetric : If the element has two sons, it is replaced by its successor or its predecessor, in each case with the same probability (or alternating). Otherwise, it is replaced by the existing son.

The tree performance is measured alternatively by the number of comparisons in a search (whether successful or not) or by the internal path length. In this work we use the latter.

2. Historical Background

The first theoretical result about deletions is due to Hibbard [5], who proved that if in a randomly built tree we delete an element, the resulting tree is also random. A similar result was encountered by Knott [7], who showed that if we randomly insert $N+K$ elements in an initially empty tree and then we delete the first K elements inserted, the result is also a random tree.

Besides, Knott proved that if we insert an element after the first deletion, the tree loses its randomness. This is intuitively explained, by regarding the leaf resulting from the deletion as one having twice the probability of the other leaves (having a ghost element inside). Also, he obtains empirical results which make him conjecture that asymmetric deletion algorithms do not worsen the behavior of a randomly built tree.

A thorough study of randomness in deletions considering different probability distributions on the tree, in particular the assumptions used herein, was done by Knuth [8]. In the case where the intervals have different probabilities due to a deletion, we still call the insertion random if we choose a new key independently from a uniform probability distribution (without loss of generality).

In 1978, Jonassen and Knuth [1] proved Knott's conjecture in trees of three elements by using a complex mathematic development in order to find the expected number of comparisons in a successful search after a great number of insertion/asymmetric-deletion pairs. The analysis method used here is similar.

Supposing that deletions preserve the randomness of the tree (i.e. in any insertion the leaves are equally likely), Mehlhorn [9] analyzed AVL trees. However, this is not a good model.

Eppinger [2] made extense simulations obtaining empirical results which show that in trees of 128 or more elements the symmetric algorithm is better than the asymmetric one, and that the latter is worse than the one of insertions only. In other words, Knott's conjecture is seemingly refuted. In particular, Eppinger's data indicate that the expected internal path length ($\overline{Ipl}$) of a tree with N elements ($N>128$) could be

$$\text{Asymmetric Algorithm} \quad 0.028\ N \log^3 N - 0.392\ N \log^2 N$$

$$\text{Symmetric Algorithm} \quad 1.22\ N \log N - 2.5\ N$$

More empirical results were obtained by Culberson [3]. In those the symmetric algorithm is again the better one ($\overline{Ipl} \approx 2N \log N$), and the best fit in the data for the $\overline{Ipl}$ in the asymmetric algorithm is $\omega(N^{3/2} \log N)$ (!!) :

$$\overline{Ipl} = 0.0869 N^{3/2} \log N + 0.1784 N^{3/2}$$

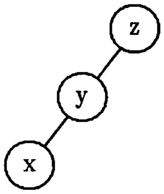
This last result would indicate that the behavior of the asymmetric algorithm is very bad. Also, Culberson made an exact analysis of trees with a small number of elements in the case where the number of keys used is the same as the number of elements in the tree after the insertion (in other words, a finite number of keys) [3]. Recently, Culberson has extended the analysis of this model, showing that the $\overline{Ipl}$ is $O(N\sqrt{N})$ in the asymmetric case [10].

3. The Analysis Method

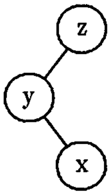
We show the method through an example: the symmetric algorithm in a three-element BST in face of insertion/deletion pairs.

The five possible three-elements BST's given $x < y < z$ are:

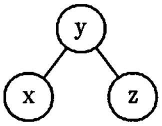
A(x,y,z)



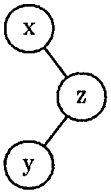
B(x,y,z)



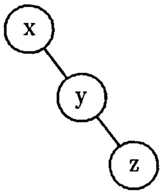
C(x,y,z)



D(x,y,z)

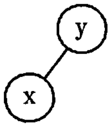


E(x,y,z)

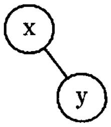


and the two possibilities with two elements (given $x < y$) are:

F(x,y)



G(x,y)



The usual insertion algorithm gives the following BST's if we insert an element z in a tree that contains x and y ($x < y$):

Initial Tree	Result if $z < x$	Result if $x < z < y$	Result if $y < z$
F(x,y)	A(z,x,y)	B(x,z,y)	C(x,y,z)
G(x,y)	C(z,x,y)	D(x,z,y)	E(x,y,z)

When deleting an element in a three-element BST by using the symmetric algorithm the following cases may occur:

Initial Tree	Delete x	Delete y	Delete z
A(x,y,z)	F(y,z)	F(x,z)	F(x,y)
B(x,y,z)	F(y,z)	F(x,z)	G(x,y)
C(x,y,z)	G(y,z)	$\frac{1}{2}$ (F(x,z) + G(x,z))	F(x,y)
D(x,y,z)	F(y,z)	G(x,z)	G(x,y)
E(x,y,z)	G(y,z)	G(x,z)	G(x,y)

In the central case above, we choose either the successor or predecessor with equal probability. To be able to study the tree's behavior, we must select a sequence of operations (insertions and/or deletions) upon it. One of the most adequate sequence of operations consist of insertion/deletion pairs because they preserve the number of elements. This allows to compare the initial and the final trees (at any given moment a search is valid, because it modifies nothing).

Then, the process to be studied is:

(i) Random Insertion of three elements

(ii) Random deletion of one element

(iii) Random Insertion of one element

(iv) Return to (ii)

If we represent an insertion by I and a deletion by D, after repeating this cycle n times, our sequence is $I\,I\,I\,(\,D\,I\,)^n\ \cdots$ and the tree's behavior depends only on the relative order of the insertions and the action of the deletion algorithm on it. One way of analysis is to consider that the $(n+3)!\ 3^n$ possible configurations (after n cycles) are equally likely. In the case $n = 1$ there exist 72 possibilities. This shows that this discrete approach is not useful.

A continuous approach is simpler. Let $f_n(x,y)dxdy$ be the differential probability that the tree is $F(X,Y)$ at the beginning of step (ii) after n elements have been deleted. Then

$$x \le X \le x + dx \quad \text{and} \quad y \le Y \le Y + dy$$

and let $g_n(x,y)dxdy$ be the corresponding probability that it is $G(X,Y)$. Let

$$a_n(x,y,z)dxdydz, \cdots \cdots, e_n(x,y,z)dxdydz$$

be the respective probabilities that the tree is $A(X,Y,Z), \ldots, E(X,Y,Z)$ at the beginning of step (iii), for some $x \le X \le x + dx$, $y \le Y \le y + dy$, $z \le Z \le z + dz$.

Now it is possible to write down recurrence relations for these differential probabilities by directly translating the algorithm into mathematical formalism. First, from the insertion algorithm we have

$$\begin{aligned} a_n(x,y,z) &= f_n(y,z) \\ b_n(x,y,z) &= f_n(x,z) \\ c_n(x,y,z) &= f_n(x,y) + g_n(y,z) \qquad \text{for } 0 \le x < y < z \le 1 \\ d_n(x,y,z) &= g_n(x,z) \\ e_n(x,y,z) &= g_n(x,y) \end{aligned} \tag{1}$$

by considering the six possible actions of step (ii). The probability is zero if $x < 0$, $x > y$, $y > z$ or $z > 1$. At the boundaries x=0, x=y, y=z, and z=1 there may be discontinuities, but, if so, they would not affect the analysis.

Secondly, from the deletion algorithm we have

$$\begin{aligned} f_{n+1}(x,y) = &\frac{1}{3}\int_0^x a_n(t,x,y) + b_n(t,x,y) + d_n(t,x,y)\ dt \\ &+ \frac{1}{3}\int_x^y a_n(x,t,y) + b_n(x,t,y) + \frac{1}{2}c_n(x,t,y)\ dt \\ &+ \frac{1}{3}\int_y^1 a_n(x,y,t) + c_n(x,y,t)\ dt\ . \end{aligned} \tag{2}$$

The equation for $g_{n+1}(x,y)$ are similar. Initially we have

$$f_0(x,y) = g_0(x,y) = 1\ ,\ \text{for } 0 \le x < y \le 1 \tag{3}$$

Now the interesting quantities are the probabilities of each tree shape at the end of n pairs. In other words

$$a_n = \int_0^1\int_0^z\int_0^y a_n(x,y,z)\ dxdydz \quad , \cdots \cdots , \quad e_n = \int_0^1\int_0^z\int_0^y e_n(x,y,z)\ dxdydz \tag{4}$$

are the probabilities that a tree of shape $A,B, \ldots ,E$ occurs after n cycles, and

$$f_n = \int_0^1\int_0^y f_n(x,y)\ dxdy \ , \ g_n = \int_0^1\int_0^y g_n(x,y)\ dxdy \tag{5}$$

the probabilities that the tree shape is F or G after n deletions and $n-1$ insertions in the loop.

To simplify these recurrences, we can look for invariant relations amongst the preceding functions. When the algorithm reaches step (iii), it is clear that the two numbers X and Y in the tree are random, except for the condition that $X < Y$. Thus we must have

$$f_n(x,y) \ + \ g_n(x,y) \ = \ 2, \text{ for } 0 \le x < y \le 1, \text{ and } n \ge 0 \ , \tag{6}$$

since the probability of $x \le X < x + dx$ and $y \le Y < y + dy$, given $X < Y$ is $2dxdy$. It is also possible to prove this relation by induction in n using equations (1), (2), and (3).

Then, using relations (1), and last equation in (2), we have a recurrence in $f_n(x,y)$ only

$$f_0(x,y) = 1$$

$$f_{n+1}(x,y) = \frac{1}{3}(\ 2 + x - y + f_n(x,y) + \frac{1}{2}\int_x^y \ f_n\ (t,y) + f_n\ (x,t) \ \ dt \ ) \quad \text{ for } n \ge 0$$

It is possible to compute now the values of f_1, f_2, etc. If the process converges for large n, the recurrence equation for f_∞ is

$$f_\infty(x,y) = 1 + \frac{x-y}{2} + \frac{1}{4}\int_x^y (f_\infty(t,y) + f_\infty(y,t) \ )\ dt \tag{7}$$

To prove that the recurrence converges, we define $r_n(x,y) = f_n(x,y) - f_\infty(x,y)$. Substracting the equations for f_n and f_∞ we have

$$r_{n+1}(x,y) = \frac{1}{3}(\ r_n(x,y) + \frac{1}{2}\int_x^y \ r_n(t,y) + r_n(x,t) \ dt \ )$$

Now, if the absolute value of r_n is bounded by α for $0 \le x < y \le 1$, we have

$$| \, r_{n+1}(x,y) \, | \le \frac{1}{3}(\ \alpha + \int_x^y \ \alpha \ dt) = \frac{1}{3} \ a \ (\ 1 + y - x \ )$$

The maximum of the bounding function above is attained in $x = 0$ and $y = 1$, i.e. $| \, r_{n+1} \, | \le \frac{2}{3}\ \alpha$. Therefore r_n converges rapidly ($O(\left(\frac{2}{3}\right)^n$)) to zero (regardless of the initial distribution) and then f_∞ exists.

The equation (7) shows a clear symmetry, and its solution is very simple, namely

$$f_\infty(x,y) \ = 1 \ \ ,$$

and is the same for all $n \ge 0$. With this the probabilities are computed directly.

Theorem 1. The probabilities for each tree shape are

$$f_n = \frac{1}{2} \quad , \quad g_n = \frac{1}{2} \ ,$$

$$a_n = b_n = d_n = e_n = \frac{1}{6} \ , \ c_n = \frac{1}{3} \ \text{ for } n \ge 0 \ .$$

Proof: By computing the probabilities using equations (1), (4) and (5). □

Then, with this algorithm, a three-element BST is always random, and its distribution equal to the initial one.

Finally, it is not necessary to compute all the probabilities because they are related. Clearly we have

$$a_n + b_n + c_n + d_n + e_n = 1 \quad \text{and} \quad f_n + g_n = 1 \quad (n \geq 0)$$

and by using equations (1), (6), and the preceding relations, it is possible to prove that

$$b_n + d_n = \frac{1}{3} \quad \text{and} \quad a_n + b_n + \frac{1}{3} - e_n = f_n \quad (n \geq 0) \quad [1].$$

4. Three-element BST's

Jonassen and Knuth [1] analize the behavior of three-element BST's, using both of the asymmetric algorithms. Their results for the stationary probabilities ($n \longrightarrow \infty$) are shown in Table I along with ours.

One way to compare these results is by looking at the probability of tree C, the balanced tree. Note that this is greater in the asymmetric algorithm, and also the clear bias towards tree F in the asymmetric cases. Intuitively, it is not clear how an asymmetry behaves better. Moreover, the exact solution to f_n is not monotonic in n and depends on Bessel functions.

Now we analize another asymmetric algorithm (which we call *particular* asymmetric). It is similar to the symmetric one, but if is possible, the predecessor is always chosen. In this case, when we delete an element this the following may occur:

Initial Tree	Delete x	Delete y	Delete z
A(x,y,z)	F(y,z)	F(x,z)	F(x,y)
B(x,y,z)	F(y,z)	F(x,z)	G(x,y)
C(x,y,z)	G(y,z)	G(x,z)	F(x,y)
D(x,y,z)	G(y,z)	G(x,z)	G(x,y)
E(x,y,z)	G(y,z)	G(x,z)	G(x,y)

By working as in the previous section, we find the recurrence equation for f_∞ to be

$$f_\infty(x,y) = 1 - y + \frac{1}{2} \int_0^y f_\infty(t,y)\, dt$$

where the right hand side is not a function of x. The solution for this equation is

$$f_\infty(x,y) = \frac{2 (1 - y)}{2 - y} .$$

The convergence is proved as in the preceding case (again $| r_{n+1} | \leq \frac{2}{3}\alpha$).

Theorem 2. The resulting probabilities are

$$f_\infty = 0.227 \quad \text{and} \quad g_\infty = 0.773$$

$$a_\infty = 0.0607 \ , \ b_\infty = 0.0607 \ , \ c_\infty = 0.379 \ , \ d_\infty = 0.2736 \ , \ e_\infty = 0.227$$

Proof: By using equations (4), (5), and the preceding solution. □

Consequently, this algorithm is better than the preceding ones, in spite of the great bias towards trees G, D, and E. Also, in this case f_n is monotonic in n. This strange behavior seems to be a transient one, and suggests that for a larger number of elements the behavior should be reversed. In the next section we present evidence supporting this conjecture.

Algorithm	a_∞	b_∞	c_∞	d_∞	e_∞	f_∞	g_∞
Asymmetric	0.150	0.196	0.352	0.137	0.164	0.516	0.484
Modified Asymmetric	0.190	0.215	0.333	0.118	0.144	0.595	0.405
Symmetric	0.167	0.167	0.333	0.167	0.167	0.5	0.5
Particular Asymmetric	0.0607	0.0607	0.379	0.2736	0.227	0.227	0.773

Table I. Stationary probabilities of the different tree shapes.

A better way to compare the algorithms consist on using the $\overline{Ipl}$ of the tree, because it is valid for any number of elements. In a three-element BST it is equivalent, because the probability for tree $\underline{C}$ ($p_c = c_n$) is linearly related to the $\overline{Ipl}$ by the equation $3(\ 1 - p_c\ ) + 2\ p_c$. Table II shows the $\overline{Ipl}$ as $n \rightarrow \infty$over the initial $\overline{Ipl}$ for all the algorithms.

Algorithm	$\overline{Ipl}(\infty)$	$\frac{\overline{Ipl}(\infty)}{\overline{Ipl}(0)}$
Asymmetric	2.648	0.9929
Modified Asymmetric	2.667	1
Symmetric	2.667	1
Particular Asymmetric	2.621	0.9828

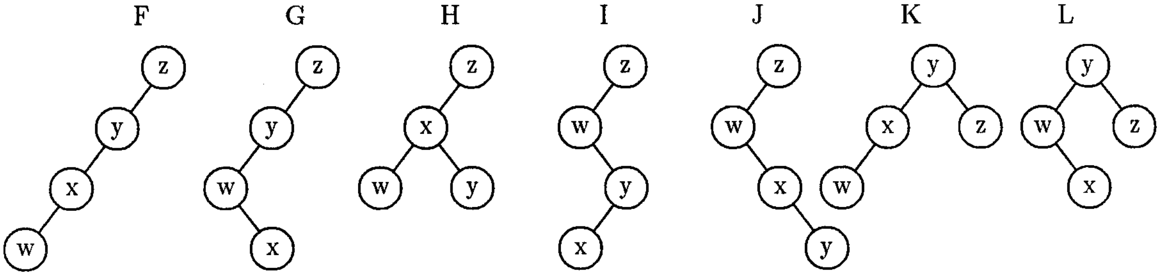
Table II. Final to Initial Expected Internal Path Length Ratio.

5. Four-element BST's.

In this section we analyze the symmetric and asymmetric algorithm in a four-element BST, for some deletion/insertion cycles. In this case we iterate the recurrences because an exact solution seems to be difficult to find. This reason will be clear when these recurrences become known. Howewer, we show that the process converges and then is possible to bound the error with respect to the steady solution.

For the iteration we used a symbolic algebraic system called MAPLE [11] in a Unix system and an ad-hoc program for the final iterations, due to the large amount of memory required. With this program the tree process for 35 insertion/deletion pairs was computed and the numerical results compared to the exact results obtained in MAPLE for the first ninth iterations. The numerical error was less than 10^{-6}. A greater number of pairs was not possible because of the size of the polynomials involved.

Here, we will go through the same steps as in the third section without making any comments. The fourteen possible four-element BST's given $w < x < y < z$ are:



and their seven symmetrical ones, which we call F', G' , . . . , and L'. Initially the probabilities of each shape after randomly inserting four elements are $\frac{1}{12}$ for H and H', $\frac{1}{8}$ for K, L, K', and L'; and $\frac{1}{24}$ for the rest. The probabilities of each type is defined the same as before, in this case computed using quadruple integrals.

When we insert an element w in a three-element BST the result is:

Initial Tree	Element w to insert is $w<x$	$x<w<y$	$y<w<z$	$z<w$
A	F	G	H	K
B	H	I	J	L
C	K	L	L’	K’
D	L’	J’	I’	H’
E	K’	H’	G’	F’
(x,y,z)	(w,x,y,z)	(x,w,y,z)	(x,y,w,z)	(x,y,z,w)

Therefore, the relations amongst the differential probabilities for this problem are:

$$\mathrm{F}_n(w,x,y,z) = \mathrm{A}_n(x,y,z)$$
$$\mathrm{G}_n(w,x,y,z) = \mathrm{A}_n(w,y,z)$$
$$\mathrm{H}_n(w,x,y,z) = \mathrm{A}_n(w,x,z) + \mathrm{B}_n(x,y,z)$$
$$\mathrm{I}_n(w,x,y,z) = \mathrm{B}_n(w,y,z)$$
$$\mathrm{J}_n(w,x,y,z) = \mathrm{B}_n(w,x,z)$$
$$\mathrm{K}_n(w,x,y,z) = \mathrm{A}_n(w,x,y) + \mathrm{C}_n(x,y,z)$$
$$\mathrm{L}_n(w,x,y,z) = \mathrm{B}_n(w,x,y) + \mathrm{C}_n(w,y,z)$$
$$\mathrm{L'}_n(w,x,y,z) = \mathrm{C}_n(w,x,z) + \mathrm{D}_n(x,y,z)$$
$$\mathrm{K'}_n(w,x,y,z) = \mathrm{C}_n(w,x,y) + \mathrm{E}_n(x,y,z)$$
$$\mathrm{J'}_n(w,x,y,z) = \mathrm{D}_n(w,y,z)$$
$$\mathrm{I'}_n(w,x,y,z) = \mathrm{D}_n(w,x,z)$$
$$\mathrm{H'}_n(w,x,y,z) = \mathrm{D}_n(w,x,y) + \mathrm{E}_n(w,y,z)$$
$$\mathrm{G'}_n(w,x,y,z) = \mathrm{E}_n(w,x,z)$$
$$\mathrm{F'}_n(w,x,y,z) = \mathrm{E}_n(w,x,y)$$

In this case the invariant relation amongst the three-element BST functions is

$$\mathrm{A}_n(x,y,z) + \mathrm{B}_n(x,y,z) + \mathrm{C}_n(x,y,z) + \mathrm{D}_n(x,y,z) + \mathrm{E}_n(x,y,z) = 6$$

When we delete an element the resulting tree is:

Algorithm	Symmetric				Asymmetric			
Initial Tree	We delete w	x	y	z	We delete w	x	y	z
F	A	A	A	A	A	A	A	A
G	A	A	A	B	A	A	A	B
H	B	$\frac{1}{2}$(A+B)	A	C	B	A	A	C
I	B	B	B	D	B	B	B	D
J	B	B	B	E	B	B	B	E
K	C	C	$\frac{1}{2}$(A+C)	A	C	C	A	A
L	C	C	$\frac{1}{2}$(B+C)	B	C	C	B	B
L’	D	$\frac{1}{2}$(D+C)	C	C	D	C	C	C
K’	E	$\frac{1}{2}$(E+C)	C	C	E	C	C	C
J’	A	D	D	D	A	D	D	D
I’	B	D	D	D	B	D	D	D
H’	C	E	$\frac{1}{2}$(E+D)	D	C	E	D	D
G’	D	E	E	E	D	E	E	E
F’	E	E	E	E	E	E	E	E
(w,x,y,z)	(x,y,z)	(w,y,z)	(w,x,z)	(w,x,y)	(x,y,z)	(w,y,z)	(w,x,z)	(w,x,y)

Then the integral equations (using the preceding ones) for the symmetric case are:

$$\begin{aligned}
\mathrm{A}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{A}_n(x,y,z) + \int_0^x \mathrm{D}_n(t,y,z)\,\mathrm{dt} + \int_0^y \mathrm{A}_n(t,y,z)\,\mathrm{dt} + \frac{1}{2}\int_x^y \mathrm{A}_n(x,t,z) + \mathrm{B}_n(y,t,z)\,\mathrm{dt} + {} \\
& \int_y^z \mathrm{A}_n(y,t,z) + \mathrm{A}_n(x,t,z) + \mathrm{B}_n(y,t,z) + \frac{1}{2}(\ \mathrm{A}_n(x,y,t) + \mathrm{C}_n(y,t,z)\ )\,\mathrm{dt} + \int_z^1 \mathrm{A}_n(y,z,t) + \mathrm{C}_n(y,z,t)\,\mathrm{dt}\ )
\end{aligned}$$

$$\begin{aligned}
\mathrm{B}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{B}_n(x,y,z) + \int_0^x \mathrm{B}_n(t,y,z) + \mathrm{A}_n(t,x,z) + \mathrm{B}_n(t,x,z) + \mathrm{D}_n(t,x,z)\,\mathrm{dt} + \frac{1}{2}\int_x^y \mathrm{A}_n(x,t,z) + \mathrm{B}_n(t,y,z)\,\mathrm{dt} + {} \\
& \int_x^z \mathrm{B}_n(x,t,z)\,\mathrm{dt} + \frac{1}{2}\int_y^z \mathrm{B}_n(x,y,t) + \mathrm{C}_n(x,t,z)\,\mathrm{dt} + \int_z^1 \mathrm{A}_n(x,z,t) + \mathrm{C}_n(x,z,t)\,\mathrm{dt}\ )
\end{aligned}$$

$$\begin{aligned}
\mathrm{C}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{C}_n(x,y,z) + \int_0^x \mathrm{E}_n(t,y,z) + \mathrm{A}_n(t,x,y) + \mathrm{B}_n(t,x,y) + \mathrm{D}_n(t,x,y)\,\mathrm{dt} + \int_0^y \mathrm{C}_n(t,y,z)\,\mathrm{dt} + {} \\
& \int_x^y \mathrm{A}_n(x,t,y) + \mathrm{B}_n(x,t,y) + \frac{1}{2}\ (\ \mathrm{D}_n(t,y,z) + \mathrm{C}_n(x,t,y) + \mathrm{E}_n(t,y,z)\ )\,\mathrm{dt} + \frac{1}{2}\int_x^z \mathrm{C}_n(x,t,z)\,\mathrm{dt} + {} \\
& \int_y^1 \mathrm{C}_n(x,y,t)\,\mathrm{dt} + \int_y^z \frac{1}{2}(\ \mathrm{A}_n(x,y,t) + \mathrm{C}_n(y,t,z) + \mathrm{B}_n(x,y,t)) + \mathrm{D}_n(y,t,z) + \mathrm{E}_n(y,t,z)\,\mathrm{dt} + {} \\
& \int_z^1 \mathrm{A}_n(x,y,t) + \mathrm{B}_n(y,z,t) + \mathrm{D}_n(y,z,t) + \mathrm{E}_n(y,z,t)\,\mathrm{dt}\ )
\end{aligned}$$

$$\begin{aligned}
\mathrm{D}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{D}_n(x,y,z) + \int_0^x \mathrm{C}_n(t,x,z) + \mathrm{E}_n(t,x,z)\,\mathrm{dt} + \int_x^z \mathrm{D}_n(x,t,z)\,\mathrm{dt} + \frac{1}{2}\int_x^y \mathrm{C}_n(x,t,z) + \mathrm{D}_n(t,y,z)\,\mathrm{dt} + {} \\
& \frac{1}{2}\int_y^z \mathrm{D}_n(x,y,t) + \mathrm{E}_n(x,t,z)\,\mathrm{dt} + \int_z^1 \mathrm{D}_n(x,y,t) + \mathrm{B}_n(x,z,t) + \mathrm{D}_n(x,z,t) + \mathrm{E}_n(x,z,t)\,\mathrm{dt}\ )
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{E}_n(x,y,z) + \int_0^x \mathrm{C}_n(t,x,y) + \mathrm{E}_n(t,x,y)\,\mathrm{dt} + {} \\
& \int_x^y \frac{1}{2}(\ \mathrm{C}_n(x,t,y) + \mathrm{E}_n(t,y,z)\ ) + \mathrm{E}_n(x,t,z) + \mathrm{E}_n(x,t,y) + \mathrm{D}_n(x,t,y)\,\mathrm{dt} + {} \\
& \frac{1}{2}\int_y^z \mathrm{D}_n(x,y,t) + \mathrm{E}_n(x,t,z)\,\mathrm{dt} + \int_y^1 \mathrm{E}_n(x,y,t)\,\mathrm{dt} + \int_z^1 \mathrm{B}_n(x,y,t)\,\mathrm{dt}\ )
\end{aligned}$$

and for the asymmetric case they are:

$$\begin{aligned}
\mathrm{A}_{n+1}(x,y,z) = {} & \frac{1}{4}(\mathrm{A}_n(x,y,z) + \int_0^x \mathrm{D}_n(t,y,z)\,\mathrm{dt} + \int_x^y \mathrm{B}_n(t,y,z)\,\mathrm{dt} + \int_x^z \mathrm{A}_n(x,t,z)\,\mathrm{dt} + {} \\
& \int_y^z \mathrm{A}_n(x,y,t) + \mathrm{A}_n(y,t,z) + \mathrm{B}_n(y,t,z) + \mathrm{C}_n(y,t,z)\,\mathrm{dt} + \int_z^1 \mathrm{A}_n(y,z,t) + \mathrm{C}_n(y,z,t)\,\mathrm{dt}\ )
\end{aligned}$$

$$\begin{aligned}\mathrm{B}_{n+1}(x,y,z) = \frac{1}{4}(&\mathrm{B}_n(x,y,z) + \int_0^x \mathrm{B}_n(t,y,z) + \mathrm{A}_n(t,x,z) + \mathrm{B}_n(t,x,z) + \mathrm{D}_n(t,x,z)\,\mathrm{dt} + \int_x^z \mathrm{B}_n(x,t,z)\,\mathrm{dt} +\\ &\int_y^z \mathrm{B}_n(x,y,t)\,\mathrm{dt} + \int_y^1 \mathrm{C}_n(x,z,t)\,\mathrm{dt} + \int_z^1 \mathrm{A}_n(x,z,t)\,\mathrm{dt}\ )\end{aligned}$$

$$\begin{aligned}\mathrm{C}_{n+1}(x,y,z) = \frac{1}{4}(&\mathrm{C}_n(x,y,z) + \int_0^x \mathrm{C}_n(t,y,z) + \mathrm{E}_n(t,y,z) + \mathrm{A}_n(t,x,y) + \mathrm{B}_n(t,x,y) + \mathrm{D}_n(t,x,y)\,\mathrm{dt} +\\ &\int_x^y \mathrm{C}_n(x,t,z) + \mathrm{C}_n(t,y,z) + \mathrm{D}_n(t,y,z) + \mathrm{E}_n(t,y,z) + \mathrm{A}_n(x,t,y) + \mathrm{B}_n(x,t,y) + \mathrm{C}_n(x,t,y)\,\mathrm{dt} +\\ &\int_y^z \mathrm{D}_n(y,t,z) + \mathrm{C}_n(x,y,t) + \mathrm{E}_n(y,t,z)\,\mathrm{dt} +\\ &\int_z^1 \mathrm{A}_n(x,y,t) + \mathrm{C}_n(x,y,t) + \mathrm{B}_n(y,z,t) + \mathrm{D}_n(y,z,t) + \mathrm{E}_n(y,z,t)\,\mathrm{dt}\ )\end{aligned}$$

$$\begin{aligned}\mathrm{D}_{n+1}(x,y,z) = \frac{1}{4}(&\mathrm{D}_n(x,y,z) + \int_0^x \mathrm{C}_n(t,x,z) + \mathrm{E}_n(t,x,z)\,\mathrm{dt} + \int_x^z \mathrm{D}_n(x,t,z)\,\mathrm{dt} +\\ &\int_y^z \mathrm{E}_n(x,t,z)\,\mathrm{dt} + \int_y^1 \mathrm{D}_n(x,y,t)\,\mathrm{dt} + \int_z^1 \mathrm{B}_n(x,z,t) + \mathrm{D}_n(x,z,t) + \mathrm{E}_n(x,z,t)\,\mathrm{dt}\ )\end{aligned}$$

$$\begin{aligned}\mathrm{E}_{n+1}(x,y,z) = \frac{1}{4}(&\mathrm{E}_n(x,y,z) + \int_0^x \mathrm{C}_n(t,x,y) + \mathrm{E}_n(t,x,y)\,\mathrm{dt} + \int_x^y \mathrm{D}_n(x,t,y) + \mathrm{E}_n(x,t,z) + \mathrm{E}_n(x,t,y)\,\mathrm{dt} +\\ &\int_y^1 \mathrm{E}_n(x,y,t)\,\mathrm{dt} + \int_z^1 \mathrm{B}_n(x,y,t)\,\mathrm{dt}\ )\end{aligned}$$

The initial conditions in both cases are:

$$A_0(x,y,z) = B_0(x,y,z) = D_0(x,y,z) = E_0(x,y,z) = 1 \ \text{ and } \ C_0(x,y,z) = 2$$

Using the invariant relation, we obtain a set of four integral equations with four unknown functions of three variables. To solve this system for each case seems to be difficult, because the order of the variables and the integration variable are never the same. Also, it seems to be impossible to develop the equations in the same way as in Jonassen and Knuth, for if that calculation was intricate, this is even more.

However, it is possible to iterate the recurrences. The first functions for the symmetric algorithm are:

$$A_1 = 1 - \frac{5}{8}y + \frac{3}{8}z \quad , \quad B_1 = 1 + \frac{1}{2}x - \frac{1}{8}y - \frac{1}{8}z \quad , \quad C_1 = 2 - \frac{1}{4}x + \frac{1}{4}z$$

$$D_1 = \frac{5}{4} + \frac{1}{8}x + \frac{1}{8}y - \frac{1}{2}z \quad , \quad E_1 = \frac{3}{4} - \frac{3}{8}x + \frac{5}{8}y$$

$$C_2 = 2 - \frac{3}{8}x + \frac{15}{32}z - \frac{5}{64}x^2 + \frac{1}{16}z\ x - \frac{3}{32}y + \frac{3}{32}y^2 - \frac{5}{64}z^2$$

and for the asymmetric one they are:

$$A_1 = 1 - \frac{1}{4}x - \frac{3}{4}y + \frac{3}{4}z \quad , \quad B_1 = 1 + \frac{3}{4}x - \frac{3}{4}y + \frac{1}{4}z \quad , \quad C_1 = 2 - x + \frac{3}{2}y - \frac{1}{2}z$$
$$D_1 = \frac{5}{4} + \frac{1}{2}x - \frac{1}{2}y - \frac{1}{4}z \quad , \quad E_1 = \frac{3}{4} + \frac{1}{2}y - \frac{1}{4}z$$

The polynomials become larger and larger every time, and we were able to compute them only up to the 35 iteration. However, it is possible to prove convergence, and then bound the error of any of the probabilities with respect the steady state.

We define $\epsilon_n^A(x,y,z) = | A_n(x,y,z) - A_\infty(x,y,z) |$. In the same way, we define ϵ_n^B, ϵ_n^C, ϵ_n^D, and ϵ_n^E. Let $\epsilon_n^A \leq \alpha$ and the same condition for ϵ_n^B, ϵ_n^D, and ϵ_n^E; and let $\epsilon_n^C \leq \beta$, with $\alpha = p\beta$ for some $p > 0$.

Let

$$r_n = \frac{a\ \epsilon_n^A + b\ \epsilon_n^B + d\ \epsilon_n^D + e\ \epsilon_n^E}{a+b+d+e} ,$$

then r_n is bounded by α, and if we bound r_{n+1} by a function of α, then to prove convergence as in previous cases, we choose adequately a,b,d, and e.

To find the best bound for the error, we choose the best linear combination and find the minimum convergence constant (< 1) by solving the problem

$$\underset{a,b,d,e \geq 0}{\text{Min}} (\text{Coef}(\underset{0<x<y<z<1}{\text{Max}} (r_{n+1}),\alpha))$$

and using the minimum in the coefficient of β in the case of more than one solution. In the symmetric case a solution to the problem is $a = e = 1$ and $b = d = \frac{5}{3}$. With this

$$r_{n+1} \leq \frac{25}{32}\alpha + \frac{\beta}{8}$$

and for prove convergence we choose $p > \frac{4}{7}$, obtaining $r_{n+1} \leq q\alpha$ with $q < 1$. Then r_n converges to zero.

Analogously, in the asymmetric case we found $a = 1$, $b = 5$, $d = 3$, $e = 7$. With this

$$r_{n+1} \leq \frac{3}{4}\alpha + \frac{3}{20}\beta$$

and we choose $p > \frac{3}{5}$ to prove convergence.

In both cases $C_n(x,y,z)$ converges by the invariant relation. Using this equation we found $\epsilon_n^C \leq \epsilon_n^A + \epsilon_n^B + \epsilon_n^D + \epsilon_n^E$. Now we express the error in the probabilities using the preceding relations. We have $\beta \leq 6$ because the maximum probability is 1, then $\alpha \leq 6p$ (if $p \geq 1$). Then in general, if we let $\epsilon(n)$ as the error in the probability, we have:

$$\epsilon(n) \leq (\ s + \frac{t}{p}\ )^n p \quad \text{with}\ \ s,\, t < 1 \ \text{ and } \ p > \frac{t}{(1-s)}$$

with $s = \frac{25}{32}$, $t = \frac{1}{8}$ in the symmetric case and $s = \frac{3}{4}$, $t = \frac{3}{20}$ in the asymmetric case. Then for example

$$a\epsilon_n^A + b\epsilon_n^B + d\epsilon_n^D + e\epsilon_n^E \leq 6\ (a+b+d+e)\ \epsilon(n)$$

and then if Δa is the error in a_n, we have

$$\Delta a \leq \frac{a+b+d+e}{a}\ \epsilon(n)$$

and similar relations for $\Delta b, \Delta d$ and Δe. The estimation error in c_n is

$$\begin{aligned}\Delta c &\leq \Delta a + \Delta b + \Delta d + \Delta e \\ &\leq a\Delta a + b\Delta b + d\Delta d + e\Delta e \quad (a,b,d,e \geq 1) \\ &\leq (a+b+d+e)\ \epsilon(n)\end{aligned}$$

Now, it is possible to find the best p which minimizes the right hand side of equation for $\epsilon(n)$, that is

$$\hat{p} = \frac{t(n-1)}{s} > \frac{t}{1-s} \quad \text{for} \quad n > \frac{1}{1-s}$$

(in the two cases $\hat{p} > 1$) and then $\epsilon(n)$ is

$$\epsilon(n) \leq \frac{nts^{n-1}}{(1-1/n)^{n-1}}$$

With this, the error of $\overline{Ipl}$ is the same error in c_n for the case that the tree have three elements. And in the case of four elements the error is proportional to $\epsilon(n)$ using the habitual error relations , because the $\overline{Ipl}$ is obtain with the formula $6 - 2(\ pk + pl + pk' + pl'\ ) - ph - ph'$ and all the preceeding probabilities is obtain with the addition of functions of the trees A, B, C, D, and E.

Again, the comparison parameter is the ratio amongst the final and initial $\overline{Ipl}$ of the algorithms. Of course, we also obtained the probability of each shape in each step, but an exhaustive list is useless. Table III shows the probability of the three-elements BST's when $n \rightarrow \infty$ using the values after 35 cycles bounded with the error in c_n (the greater).

Algorithm	a_∞	b_∞	c_∞	d_∞	e_∞
Symmetric (±0.014)	0.150	0.158	0.384	0.158	0.150
Asymmetric (±0.016)	0.226	0.208	0.367	0.100	0.098
Initial Value	0.166	0.166	0.333	0.166	0.166

Table III. Asymptotic probabilities of the three-elements BST's.

Table IV shows the four-element BST steady probabilities obtained bounding the values at the end of 35 cycles.

In spite of the fact that some probabilities of balanced trees are greater in the asymmetric case, the $\overline{Ipl}$ is worse. Table V shows the ratio $\frac{\overline{Ipl}(k)}{\overline{Ipl}(0)}$, where $\overline{Ipl}(k)$ denotes the $\overline{Ipl}$ after k cycles, at end of step (ii) (three elements) and at end of step (iii) (four elements) of the process.

The small variation in the last terms of the above results, allows to verify the trend. The symmetric algorithm converges near 0.98 and the asymmetric one near 0.985. So far, the symmetric algorithm is the better one, and both improve over the initial tree. This is in accordance with Eppinger's results that show the same for small n. Then, in fact that $\overline{Ipl}(0)$ is a constant the preceding ratio when $n \rightarrow \infty$ is bounded by the values of Table VI.

Figure 1 shows the difference more clearly.

6. Conclusions

An analysis for five-element BST's needs the 52 possible trees of this size, which entails 13 integral equations, with 13 unknown functions of four variables. Then, it is clear that is necessary to find a better method of analisys for the case in which deletions exist. Then, a partial or exact analysis for large trees is virtually impossible in this way.

These results are the first in accordance with the empirical data obtained recently [2,3] and shows that at least for $n = 4$ the symmetric deletion algorithm is better. Therefore, we conjecture that the case of $n = 3$ is a transient, and for large n the behavior is very good for the symmetric algorithm and poor for the asymmetric one (see [10]). Also, this work shows that if an

Tree	Initial Value	Symmetric Algorithm (±0.005)	Asymmetric Algorithm (±0.006)
F	0.0416	0.033	0.046
G	0.0416	0.034	0.055
H	0.0833	0.084	0.124
I	0.0416	0.036	0.047
J	0.0416	0.039	0.058
K	0.1250	0.134	0.145
L	0.1250	0.140	0.160
L'	0.1250	0.140	0.108
K'	0.1250	0.134	0.121
J'	0.0416	0.039	0.018
I'	0.0416	0.036	0.022
H'	0.0833	0.084	0.050
G'	0.0416	0.034	0.021
F'	0.0416	0.033	0.025

Table IV. Asymptotic probabilities of four-elements BST's.

Algorithm	Symmetric		Asymmetric	
Number of Elements	3	4	3	4
Cycle				
0	1	1	1	1
1	0.99218	0.99181	0.99218	0.99181
2	0.98769	0.98708	0.98886	0.98793
3	0.98502	0.98427	0.98754	0.98611
4	0.98339	0.98256	0.98709	0.98529
5	0.98240	0.98152	0.98701	0.98496
6	0.98179	0.98089	0.98706	0.98484
7	0.98141	0.98051	0.98715	0.98481
8	0.98119	0.98028	0.98722	0.98481
9	0.98105	0.98015	0.98727	0.98482
10	0.98097	0.98007	0.98731	0.98483
11	0.98092	0.98001	0.98732	0.98483
12	0.98089	0.97999	0.98733	0.98483
13	0.98088	0.97997	0.98734	0.98483
14	0.98087	0.97996	0.98734	0.98483
15	0.98086	0.97996	0.98734	0.98483
20	0.98085	0.97995	0.98733	0.98482
35	0.98085	0.97995	0.98733	0.98482

Table V. Final to Initial $\overline{Ipl}$ Ratio after k cycles.

exact analysis is complicated, a numerical analysis is a good tool for trying to solve a problem, bounding the error of the values.

Acknowledgments

The author wishes to acknowledge the helpful comments of Patricio Poblete.

Algorithm	Symmetric	Asymmetric
Elements in the tree	$\frac{\overline{Ipl}(\infty)}{\overline{Ipl}(0)}$	$\frac{\overline{Ipl}(\infty)}{\overline{Ipl}(0)}$
3	0.981 ± 0.005	0.987 ± 0.004
4	0.980 ± 0.004	0.9848 ± 0.0003

Table VI. Final ($n \rightarrow \infty$) to Initial $\overline{Ipl}$ ratio.

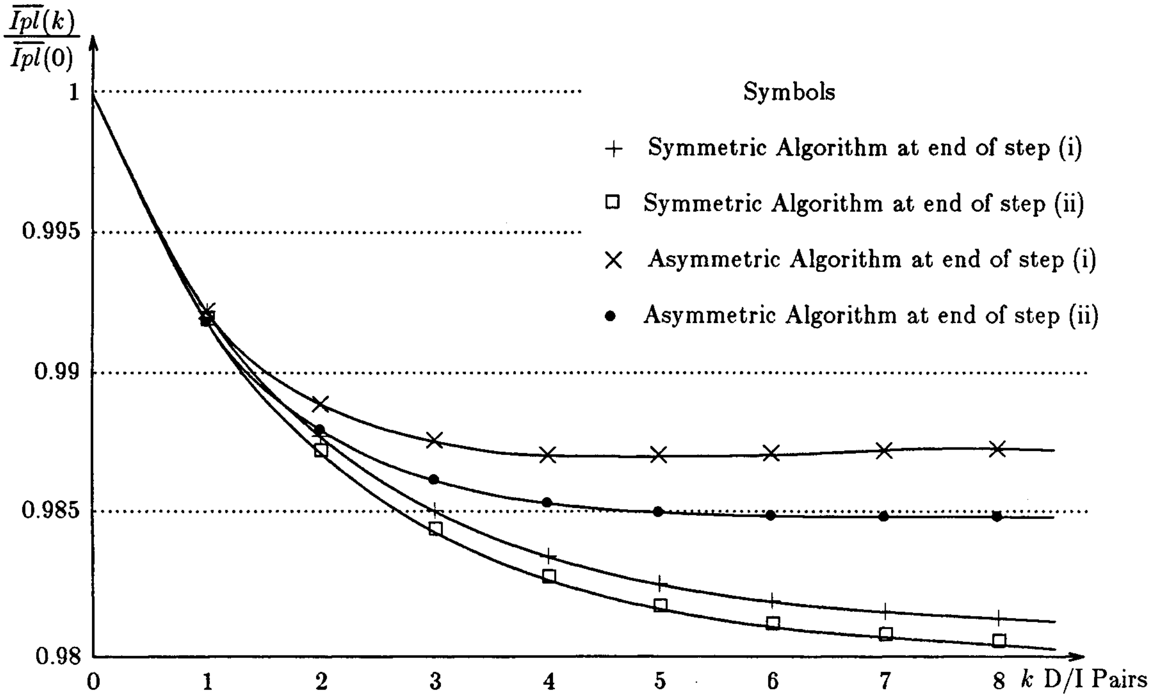


Figure 1. Ratio $\frac{\overline{Ipl}(k)}{\overline{Ipl}(0)}$ for k Deletion/Insertion Pairs

References

1. Jonassen, A.T. y Knuth, D.E. A Trivial Algorithm Whose Analysis Isn't, *Journal of Computer and System Science* **16**, 13(June 1978), 301-322.
2. Eppinger, J.L. An Empirical Study of Insertion and Deletion in Binary Trees, *Communications of the ACM* **26**, 9(September 1983), 663-669.
3. Culberson, J.C. "Updating Binary Trees", M.S. Thesis, Report CS-84-08, Department of Computer Science, University of Waterloo, Waterloo, Canada, March 1984.
4. Baeza-Yates, R.A. "Análisis de Algoritmos en Arboles de Búsqueda" (Analysis of Algorithms in Search Trees), M.S. Thesis, Department of Computer Science, University of Chile, Santiago, Chile, January 1985.

5. Hibbard, T.N. "Some Combinatorial Properties of Certain Trees with Applications to Searching and Sorting", *Journal of ACM* 9, 1(Jan 1962), 13-28.

6. Knuth, D.E. "The Art of Computer Programming", Vol. 3, Reading, Mass.; Addison-Wesley, 1973.

7. Knott, G.D. "Deletions in Binary Storage Trees", Ph.D. Thesis, Computer Science Department, Stanford University, Report STAN-CS-75-491, May 1975.

8. Knuth, D.E. "Deletions That Preserve Randomness", *IEEE Transactions on Software Engineering* **3**, 5(September 1977), 351-359.

9. Mehlhorn, K. A Partial Analysis of Height-Balanced Trees under Random Insertions and Deletions, *SIAM Journal of Computing* **11**, (November 1979) , 748-760.

10. Culberson, J. The Effect of Asymmetric Deletions on Binary Search Trees, Ph.D. Thesis, Report CS-86-15, Dept. of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, May 1986.

11. Geddes, K.O., Gonnet, G.H y Char B.W. "MAPLE User's Manual, Second Edition", Report CS-82-40, Department of Computer Science, University of Waterloo, Waterloo, Canada, December 1982.