The Analysis of a Fringe Heuristic for Binary Search Trees

Patricio V. Poblete †

J. Ian Munro

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-82-22
July 1982

ABSTRACT

We present a detailed analysis for the behaviour of binary search trees built by using a heuristic that performs only local reorganizations at the bottom of the tree.

† On leave from the University of Chile.

The Analysis of a Fringe Heuristic for Binary Search Trees

Patricio V. Poblete

J. Ian Munro

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

1. Introduction

Binary search trees constitute a well known data structure for efficient retrieval and modification of information. If elements are inserted in random order by simply appending them at the external nodes at which searches for them are terminated, then the mean and the variance of the number of probes required for a search are both about $2\ln N$, where N is the number of elements in the tree [5]. There is, of course, the danger that such trees may degenerate into linear lists. One approach to avoiding drastically unbalanced structures is the introduction of rigid balance disciplines such as height [1] or weight balance [6]. Such schemes guarantee logarithmic search and update costs, but do add to space requirements and coding difficulty. In this paper we suggest a class of simple heuristics for inserting elements into a tree such that drastically unbalanced trees are much less likely to occur than under the naive scheme; furthermore, the expected search time is also reduced.

The basic idea is very simple: whenever a son is appended to a node which itself is an only son, a rotation of the three nodes is performed to place the median of the three values at the root of the subtree with the other two as sons (see Figure 1). No other balancing action is taken and so we refer to a technique of this form as a *fringe* heuristic.

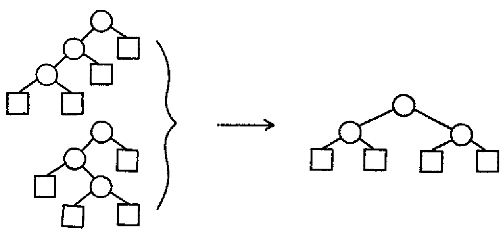


Figure 1
Basic fringe heuristic
○ denotes an internal node or data value
□ denotes an external node or missing element

The origins of this heuristic can be traced to the work of Bell[2], and Walker and Wood[8]. Further analysis was done by Itai and Rodeh [4]. Walker and Wood describe a class of trees called "k-locally balanced trees," that coincide with the class of trees generated by this heuristic for $k=1$. However, they provide only a superficial analysis, and empirical results. We will introduce later a generalization of our heuristic that differs from theirs. The heuristic is also implicit in some of the work on analysis of balanced trees. For instance, Guibas and Sedgewick[3] analyze the average number of comparisons for an AVL tree, assuming all rotations occur only at the "bottom" of the tree. Under that assumption, AVL trees become the trees generated by the heuristic. There is also an analogy between this class of schemes and Quicksort[7]. We observe, however, that although the analysis of the mean carries over, the higher moments do not.

We are able to produce the generating function for the distribution of search paths under this scheme. In principle, this generating function can be extended to any scheme that chooses the median of an odd number of elements at the fringe to become the root of the subtree containing these elements. However, to obtain this generating function we must determine the (exact) eigenvalues of a matrix whose size grows as the number of elements chosen becomes larger. This can be very complicated. For matrices larger than 4x4 it involves finding polynomial roots which, of course, need not be algebraic, and appear to have no (helpful) special structure. However, we can determine asymptotic expressions for the mean and the variance of the distribution without finding the full generating function, thus providing an analysis for the general case.

2. Analysis of the Basic Heuristic

Let N be the number of data values or internal nodes in a binary search tree, and n be the number of termination points for unsuccessful searches or external nodes ($n=N+1$). Let $P_{n,k}$ be the probability that a k comparisons are needed for an unsuccessful search, and let $P_n(z)$ be the generating function

$$P_n(z) = \sum_{k\geq 0} P_{n,k} z^k$$

The main complexity measures for binary search tree are the average search time and its variance. We denote C_N and C_N' the average number of comparisons needed in a successful and in an unsuccessful search, respectively. Similarly, V_N and V_N' are the associated variances.

We will be primarily concerned with obtaining C_N' and V_N'. The following two relations can be used to derive the other quantities:

$$C_N = (1+\frac{1}{N})C_N' - 1$$

$$V_N = (1+\frac{1}{N})V_N' - \frac{1}{N}(1+\frac{1}{N})(C_N')^2 + 2$$

For naively formed binary search trees, the following results are well known [5]:

$$P_n(z) = (2z+n-2)(2z+n-3)\cdots(2z)/n!$$

$$C_N' = 2(H_{N+1}-1) \approx (2\ln 2)\lg N$$

$$V_N' = 2H_{N+1} - 4H_{N+1}^{(2)} + 2 \approx (2\ln 2)\lg N$$

where lg denotes the base 2 logarithm, $H_N^{(i)}$ denotes $\sum_{1\leq j\leq N} \frac{1}{j^i}$ and $H_N \equiv H_N^{(1)}$.

To begin our analysis, it is convenient to assign levels to the tree, starting from the root (the root is at level 0). We make the assumption that the n

external nodes are all equally likely to be hit by unsuccessful searches or by insertions. Then $P_{n,k}$ is the probability that the node chosen is at level k. More precisely,

$$P_{n,k} = \frac{\text{average number of external nodes at level } k}{n}$$

For the usual insertion method for binary search trees it is easy to find a recurrence relating $P_{n+1,k}$ to $P_{n,k}$ and $P_{n,k-1}$. This is not possible when we use the heuristic, but we can break $P_{n,k}$ into three components, for which we are able to find recurrences. Let us say that an external node is of type A if it is attached to a single node, of type B if it is attached to the bottom of a pair, and of type C if it is attached to the top of a pair (see Figure 2).

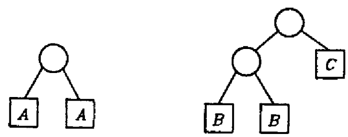


Figure 2
Types of external nodes

Let us define

$$A_{n,k} = \frac{\text{average number of } A \text{ nodes at level } k}{n}$$

$$B_{n,k} = \frac{\text{average number of } B \text{ nodes at level } k}{n}$$

$$C_{n,k} = \frac{\text{average number of } C \text{ nodes at level } k}{n}$$

Clearly we have

$$P_{n,k} = A_{n,k} + B_{n,k} + C_{n,k}$$

and also

$$C_{n,k} = \frac{1}{2} B_{n,k+1}$$

Let us try to find a recurrence relation for these quantities. Suppose we insert a new element. There are two possible transitions, shown in Figure 3.

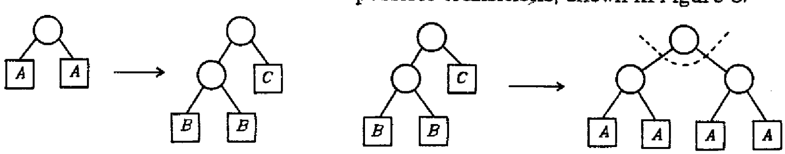


Figure 3
Transformation rules

Therefore,

$$A_{n+1,k} = \frac{1}{n+1}\left[nA_{n,k} + 4(B_{n,k}+C_{n,k-1}) - 2A_{n,k}\right]$$

$$B_{n+1,k} = \frac{1}{n+1}[nB_{n,k} + 2A_{n,k-1} - 2(B_{n,k}+C_{n,k-1})]$$

$$C_{n+1,k} = \frac{1}{n+1}[nC_{n,k} + A_{n,k} - (B_{n,k+1}+C_{n,k})]$$

Using the relationship between B and C, we can eliminate C, and rewrite the resulting equations as follows:

$$A_{n+1,k} = A_{n,k} + \frac{1}{n+1}[-3A_{n,k} + 6B_{n,k}]$$

$$B_{n+1,k} = B_{n,k} + \frac{1}{n+1}[2A_{n,k-1} - 4B_{n,k}]$$

with the boundary conditions

$$A_{2,1} = 1$$

$$A_{2,k} = 0 \quad \text{for } k \neq 1$$

$$B_{2,k} = 0 \quad \text{for all } k$$

We now introduce the generating functions $A_n(z)$, $B_n(z)$, and $C_n(z)$, defined analogously to $P_n(z)$. Note that

$$C_n(z) = \frac{1}{2z}B_n(z)$$

and therefore

$$P_n(z) = A_n(z) + (1+\frac{1}{2z})B_n(z)$$

Using these generating functions and writing in matrix notation, we get the recurrence

$$\begin{pmatrix} A_{n+1}(z) \\ B_{n+1}(z) \end{pmatrix} = (I + \frac{1}{n+1}\begin{pmatrix} -3 & 6 \\ 2z & 4 \end{pmatrix}) \begin{pmatrix} A_n(z) \\ B_n(z) \end{pmatrix}$$

$$\begin{pmatrix} A_2(z) \\ B_2(z) \end{pmatrix} = \begin{pmatrix} z \\ 0 \end{pmatrix}$$

Lemma 2.1

Let $\vec{A}_n(z)$ be a t dimensional vector that satisfies the recurrence

$$\vec{A}_{n+1}(z) = (I + \frac{1}{n+1}H(z))\vec{A}_n(z)$$

given $\vec{A}_t(z)$. If $H(z)$ has distinct eigenvalues $\lambda_1(z), \ldots, \lambda_t(z)$, then

$$\vec{A}_n(z) = E(z)\ \mathrm{diag}(\pi_t^n(\lambda_1(z)), \ldots, \pi_t^n(\lambda_t(z)))\ E^{-1}(z)$$

where $E(z)$ is the matrix whose columns are the eigenvectors associated with $\lambda_1(z), \ldots, \lambda_t(z)$, and $\pi_t^n(\lambda)$ is the function

$$\pi_t^n(\lambda) = \prod_{t<j\le n} \frac{j+\lambda}{j}$$

Proof

The solution of the recurrence is

$$\vec{A}_n(z) = \prod_{t<j\le n} \left(\frac{jI+H(z)}{j}\right) \vec{A}_t(z)$$

By definition of $E(z)$, we have $E^{-1}(z)H(z)E(z) = D(z)$, where $D(z) = \mathrm{diag}(\lambda_1(z), \ldots, \lambda_t(z))$. Then

$$\prod_{t<j\le n} (jI+H(z)) = \prod_{t<j\le n} (jE(z)E^{-1}(z) + E(z)D(z)E^{-1}(z))$$
$$= E(z)\left(\textstyle\prod_{t<j\le n}(jI+D(z))\right) E^{-1}(z)$$

This is now easy to compute, because

$$\prod_{t<j\le n} (jI+D(z)) = \mathrm{diag}\Big(\prod_{t<j\le n} (j+\lambda_1(z)), \ldots, \prod_{t<j\le n} (j+\lambda_t(z))\Big) \quad \blacksquare$$

We can apply this result to our particular problem. The eigenvalues of matrix $H(z)$ are

$$\lambda_1 = -\frac{7}{2} + \frac{w}{2}, \qquad \lambda_2 = -\frac{7}{2} - \frac{w}{2}$$

where $w = \sqrt{1+48z}$, and

$$E(z) = \begin{pmatrix} 1 & 1 \\ -\frac{w-1}{12} & -\frac{w+1}{12} \end{pmatrix} \qquad E^{-1}(z) = \frac{6}{w}\begin{pmatrix} \frac{w+1}{12} & 1 \\ \frac{w-1}{12} & -1 \end{pmatrix}$$

The solution can be written as

$$\begin{pmatrix} A_n(z) \\ B_n(z) \end{pmatrix} = \frac{z}{2w}\begin{pmatrix} (\pi^{(1)}+\pi^{(2)})w + (\pi^{(1)}-\pi^{(2)}) \\ 4z(\pi^{(1)}-\pi^{(2)}) \end{pmatrix}$$

where

$$\pi^{(1)} = \pi_t^n\left(-\frac{7}{2}+\frac{w}{2}\right), \qquad \pi^{(2)} = \pi_t^n\left(-\frac{7}{2}-\frac{w}{2}\right)$$

From this we can compute the generating function

$$P_n(z) = \frac{z(w+4z+3)}{2w}\pi^{(1)} - \frac{z(w-4z-3)}{2w}\pi^{(2)}$$

Now it is routine task to compute the mean and the variance by differentiation. We have therefore proved:

Theorem 2.1

$$C_N' = \frac{12}{7}H_{N+1} - \frac{75}{49}$$

for $N \ge 6$, and

$$V_N' = \frac{300}{343}H_{N+1} - \frac{144}{49}H_{N+1}^{(2)} + \frac{5056}{2401} + \frac{2304}{343}\,\frac{1}{(N+1)\cdot N\cdot(N-1)\cdot\cdots\cdot(N-5)}$$

for $N \ge 13$. $\blacksquare$

Proof

The solution of the recurrence is

$$\vec{A}_n(z) = \prod_{t<j\le n} \left(\frac{jI+H(z)}{j}\right) \vec{A}_t(z)$$

By definition of $E(z)$, we have $E^{-1}(z)H(z)E(z) = D(z)$, where $D(z) = \mathrm{diag}(\lambda_1(z), \ldots, \lambda_t(z))$. Then

$$\prod_{t<j\le n} (jI+H(z)) = \prod_{t<j\le n} (jE(z)E^{-1}(z) + E(z)D(z)E^{-1}(z))$$
$$= E(z)\left(\prod\nolimits_{t<j\le n}(jI+D(z))\right) E^{-1}(z)$$

This is now easy to compute, because

$$\prod_{t<j\le n} (jI+D(z)) = \mathrm{diag}\left(\prod_{t<j\le n} (j+\lambda_1(z)), \ldots, \prod_{t<j\le n} (j+\lambda_t(z))\right) \quad \blacksquare$$

We can apply this result to our particular problem. The eigenvalues of matrix $H(z)$ are

$$\lambda_1 = -\frac{7}{2}+\frac{w}{2}, \qquad \lambda_2 = -\frac{7}{2}-\frac{w}{2}$$

where $w = \sqrt{1+48z}$, and

$$E(z) = \begin{pmatrix} 1 & 1 \\ -\frac{w-1}{12} & -\frac{w+1}{12} \end{pmatrix} \qquad E^{-1}(z) = \frac{6}{w}\begin{pmatrix} \frac{w+1}{12} & 1 \\ \frac{w-1}{12} & -1 \end{pmatrix}$$

The solution can be written as

$$\begin{pmatrix} A_n(z) \\ B_n(z) \end{pmatrix} = \frac{z}{2w}\begin{pmatrix} (\pi^{(1)}+\pi^{(2)})w + (\pi^{(1)}-\pi^{(2)}) \\ 4z\,(\pi^{(1)}-\pi^{(2)}) \end{pmatrix}$$

where

$$\pi^{(1)} = \pi_t^n\left(-\frac{7}{2}+\frac{w}{2}\right), \qquad \pi^{(2)} = \pi_t^n\left(-\frac{7}{2}-\frac{w}{2}\right)$$

From this we can compute the generating function

$$P_n(z) = \frac{z\,(w+4z+3)}{2w}\pi^{(1)} - \frac{z\,(w-4z-3)}{2w}\pi^{(2)}$$

Now it is routine task to compute the mean and the variance by differentiation. We have therefore proved:

Theorem 2.1

$$C_N' = \frac{12}{7}H_{N+1} - \frac{75}{49}$$

for $N\ge 6$, and

$$V_N' = \frac{300}{343}H_{N+1} - \frac{144}{49}H_{N+1}^{(2)} + \frac{5056}{2401} + \frac{2304}{343}\,\frac{1}{(N+1)\cdot N\cdot(N-1)\cdot\cdots\cdot(N-5)}$$

for $N\ge 13$. $\blacksquare$

The significance of this result is best understood by comparing it with the behaviour of naively formed thees and with perfectly balanced trees. We have that $C_N' \approx 1.19 \lg N$; in other words, this scheme lies almost exactly halfway between perfectly balanced trees and naively formed trees. The simple fringe heuristic seems a small price for eliminating half the "waste" of the naively formed structures. Perhaps of greater significance is the variance, which is reduced by more of a factor of 2, from roughly $1.4 \lg N$ to about $.6 \lg N$. This is indicative of the fact (attested by more detailed analysis of the generating function) that the probability of "very bad" trees is dramatically reduced.

It is quickly noted that the worst case behaviour of our scheme leads to a tree of height $\lceil (N+1)/2 \rceil$, rather than N if the elements are presented in order. Although still linear, this is clearly an improvement and, interestingly, the "waste" is again reduced by a factor of 2. In fairness we should observe that there are sequences of insertions for which our scheme produces a substantially worse tree than the naive method. As indicated in the example below, there exists a sequence of N insertions that generates a tree of height $\Theta(\sqrt{N})$ under the naive method and a tree of height $\Theta(N)$ when the heuristic is applied. However, the fact that both the average search time and its variance decrease tells us that these anomalous situations must happen with very small probability.

We define a family of trees $T_h(p)$ $(h,p \geq 1)$, each built by performing normal insertions on an initially empty tree. The sequence of keys inserted is $K_p, K_{p+1}, \ldots,$ and the order relation among them is such that the resulting trees have the shapes indicated in Figure 4 (for convenience we write j instead of K_j).

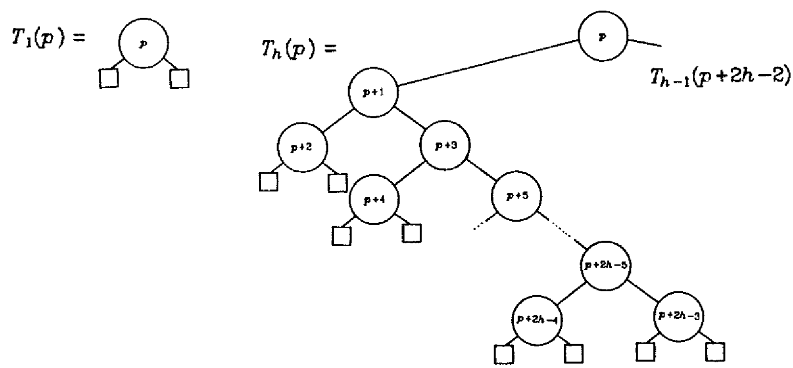


Figure 4
Trees built using the naive method

It is easy to see that a tree $T_h(p)$ has height h and $h(h-1)+1$ internal nodes. Let $T_h'(p)$ be the tree obtained by processing the same sequence of insertions using the heuristic. The resulting family of trees is shown in Figure 5.

The height of a tree $T_h'(p)$ is $h(h-1)/2$. Therefore, we see that a sequence of insertions that creates a tree of height h using the naive method, creates a tree of height $\Theta(h^2)$ when the heuristic is used. In terms of the number of nodes, the height goes from $\Theta(\sqrt{N})$ to $\Theta(N)$. Similar results hold for the expected search time.

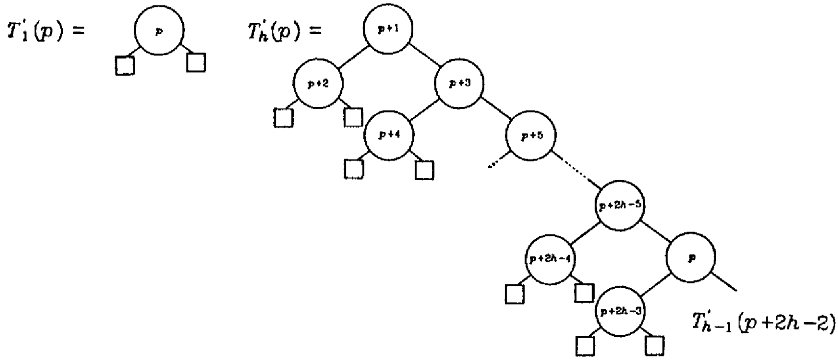


Figure 5
Trees built using the heuristic

3. A Generalized Heuristic

The simple heuristic we have just analyzed can be described as waiting until three elements have been inserted in a subtree before deciding which one will become the root of that subtree. This process is illustrated in Figure 6.

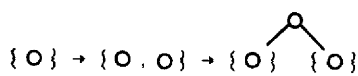


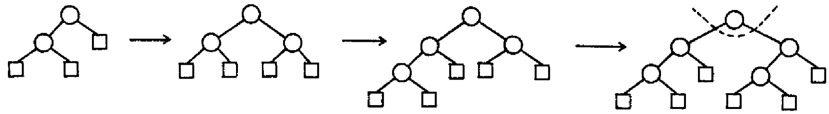
Figure 6

This has the obvious generalization of accumulating some odd number of elements, say $2t-1$, and then letting the median of the set become the root of the subtree. Figure 7 shows this process for $t=3$.

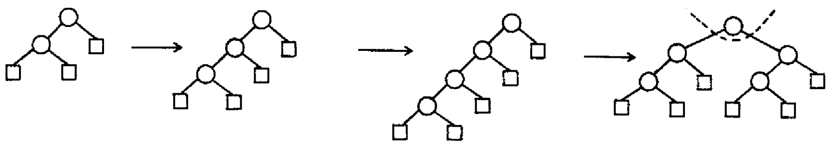
Figure 7

How the elements inside a set are organized for retrieval is not particularly important from the asymptotic point of view, because that can only add a constant amount to the search time. Two extreme alternatives are illustrated in Figure 8(a) and (b).

We will assume that some sequence of trees $T^{(t)}, \ldots, T^{(2t-1)}$ has been chosen, such that $T^{(j)}$ has j external nodes, for $t \le j < 2t$. The algorithm starts



(a) A scheme that minimizes search time



(b) A scheme that maximizes search time

Figure 8

with one instance of $T^{(t)}$, advances to the next tree in the sequence with each insertion, and closes the cycle by constructing a tree that has a root and two instances of trees $T^{(t)}$ as its left and right subtrees. We will denote by $h^{(j)}$ the height of tree $T^{(j)}$, and by $[e_0^{(j)}, e_1^{(j)}, \cdots]$ its number of external nodes in each level, starting from the bottom.

For instance, from Figure 8(a) we have

$h^{(3)}=3$; $[2, 1, 0]$

$h^{(4)}=3$; $[4, 0, 0]$

$h^{(5)}=4$; $[2, 3, 0, 0]$

and from Figure 8(b) we have

$h^{(3)}=3$; $[2, 1, 0]$

$h^{(4)}=4$; $[2, 1, 1, 0]$

$h^{(5)}=5$; $[2, 1, 1, 1, 0]$

Let $A_{n,k}^{(j)}$ be the probability that a random external node is in level k of the whole tree, and is attached to the *bottom* of a tree $T^{(j)}$. Let $A_n^{(j)}(z)$ be the corresponding generating function. We then have

$$P_{n,k} = \sum_{t \le j < 2t} \sum_{0 \le i < h^{(j)}} \frac{e_i^{(j)}}{e_0^{(j)}} A_{n,k+i}^{(j)}$$

or

$$P_n(z) = \sum_{t \le j < 2t} \Big(\sum_{0 \le i < h^{(j)}} \frac{e_i^{(j)}}{e_0^{(j)}} \frac{1}{z^i} \Big) A_n^{(j)}(z)$$

It is not hard to obtain a recurrence for the $A_{n,k}^{(j)}$. If $t < j < 2t$, we have

$$A_{n+1,k}^{(j)} = A_{n,k}^{(j)} + \frac{1}{n+1}\Big[-(j+1)A_{n,k}^{(j)} + (j-1)\frac{e_0^{(j)}}{e_0^{(j-1)}} A_{n,k+h^{(j-1)}-h^{(j)}}^{(j-1)}\Big]$$

and for $j=t$, we have

$$A_{n+1,k}^{(t)} = A_{n,k}^{(t)} + \frac{1}{n+1}\left[-(t+1)A_{n,k}^{(t)} + 2(2t-1)\frac{e_0^{(t)}}{e_0^{(2t-1)}}A_{n,k+h^{(2t-1)}-h^{(t)}-1}^{(2t-1)}\right]$$

In terms of generating functions, these recurrences become

$$A_{n+1}^{(j)}(z) = A_n^{(j)}(z) + \frac{1}{n+1}\left[-(j+1)A_n^{(j)}(z) + (j-1)\frac{e_0^{(j)}}{e_0^{(j-1)}}z^{h^{(j)}-h^{(j-1)}}A_n^{(j-1)}(z)\right]$$

for $t<j<2t$, and

$$A_{n+1}^{(t)}(z) = A_n^{(t)}(z) + \frac{1}{n+1}\left[-(t+1)A_n^{(t)}(z) + 2(2t-1)\frac{e_0^{(t)}}{e_0^{(2t-1)}}z^{h^{(t)}-h^{(2t-1)}+1}A_n^{(2t-1)}(z)\right]$$

This can be rewritten in matrix notation as

$$\vec{A}_{n+1}(z) = (I+\frac{1}{n+1}H(z))\vec{A}_n(z)$$

where $\vec{A}_n(z) = (A_n^{(t)}(z), \ldots, A_n^{(2t-1)}(z))^T$, and where the matrix $H(z)$ is shown in Figure 9.

$$\begin{bmatrix} -(t+1) & 0 & 0 & \cdots & 2(2t-1)\frac{e_0^{(t)}}{e_0^{(2t-1)}}z^{h^{(t)}-h^{(2t-1)}+1} \\ t\frac{e_0^{(t+1)}}{e_0^{(t)}}z^{h^{(t+1)}-h^{(t)}} & -(t+2) & 0 & \cdots & 0 \\ 0 & (t+1)\frac{e_0^{(t+2)}}{e_0^{(t+1)}}z^{h^{(t+2)}-h^{(t+1)}} & -(t+3) & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & -(2t) \end{bmatrix}$$

Figure 9
The matrix $H(z)$

Lemma 3.1

$$\det(H(z)-\lambda I) = (-1)^t[(\lambda+t+1)(\lambda+t+2)\cdots(\lambda+2t) - (t+1)(t+2)\cdots(2t)]$$

Proof

Expand the determinant by the first row. ■

As an illustration of the preceding discussion, for the sequence of trees in Figure 8(a) we have

$$P_n(z) = (1+\frac{1}{2z})A_n^{(3)}(z) + A_n^{(4)}(z) + (1+\frac{3}{2z})A_n^{(5)}(z)$$

$$H(z) = \begin{bmatrix} -4 & 0 & 10 \\ 6 & -5 & 0 \\ 0 & 2z & -6 \end{bmatrix}$$

$$\det(H(z)-\lambda I) = -[(\lambda+4)(\lambda+5)(\lambda+6)-4\cdot 5\cdot 6\cdot z]$$

and, for the trees in Figure 8(b) we have

$$P_n(z) = (1+\frac{1}{2z})A_n^{(3)}(z) + (1+\frac{1}{2z}+\frac{1}{2z^2})A_n^{(4)}(z) + (1+\frac{1}{2z}+\frac{1}{2z^2}+\frac{1}{2z^3})A_n^{(5)}(z)$$

$$H(z) = \begin{bmatrix} -4 & 0 & 10/z \\ 3z & -5 & 0 \\ 0 & 4z & -6 \end{bmatrix}$$

$$\det(H(z)-\lambda I) = -[(\lambda+4)(\lambda+5)(\lambda+6)-4\cdot 5\cdot 6\cdot z]$$

We will denote

$$p(\lambda,z) = (\lambda+t+1)\cdots(\lambda+2t) - (t+1)\cdots(2t)z$$

and $p(\lambda) = p(\lambda,1)$.

Lemma 3.2

All the roots of $p(\lambda)$ are pairwise distinct, one of them is zero and all the others have strictly negative real parts.

Proof

The proof is a slight modification of [5, ex. 6.2.4-10]. Let λ be a root with multiplicity two or more. Then $p(\lambda)=0$ and also $p'(\lambda)=0$. But $p'(\lambda)=0$ implies that

$$\frac{1}{\lambda+t+1}+\cdots+\frac{1}{\lambda+2t}=0$$

This can only happen if λ is real and $-2t<\lambda<-(t+1)$. But this implies

$$|\lambda+t+1|\cdots|\lambda+2t| < (t+1)\cdots(2t)$$

a contradiction with $p(\lambda)=0$.

Let $\lambda_1,\ldots,\lambda_t$ be the roots of $p(\lambda)=0$. Now we observe that $|\lambda_k+2t| \le 2t$ for all k, since otherwise we would have

$$|\lambda_k+t+1|\cdots|\lambda_k+2t| > (t+1)\cdots(2t)$$

Clearly $p(\lambda)$ has zero as a root. If we take $\lambda_1=0$, then we must have Re $\lambda_k<0$ for $2\le k\le t$. ■

We will denote $\lambda_1(z),\ldots,\lambda_t(z)$ the roots of $p(\lambda,z)=0$, with the convention that $\lambda_k(1)=\lambda_k$ for all k, where the λ_k are the roots of $p(\lambda)=0$, as defined in the proof of the preceding lemma. In particular, $\lambda_1(1)=0$.

Lemma 3.3

There exist functions $\alpha_1(z),\ldots,\alpha_t(z)$ that do not depend on n such that

$$P_n(z) = \sum_{1\le k\le t} \alpha_k(z)\, \pi_t^n(\lambda_k(z))$$

Proof

We know that $P_n(z)$ is a linear combination of the $A_n^{(k)}(z)$, with coefficients that do not depend on n. The results then follows by direct application of Lemma 2.1. ■

Lemma 3.4

Let $\alpha_1(z), \ldots, \alpha_t(z)$ be defined as before. Then $\alpha_1(1)=1$ and $\alpha_k(1)=0$ for $2 \le k \le t$.

Proof

By definition of the Gamma function and recalling our definition of $\pi_t^n(\lambda)$,

$$\pi_t^n(\lambda) \sim \frac{\Gamma(t+1)}{\Gamma(\lambda+t+1)} n^{\lambda}$$

Since $P_n(z)$ is a probability generating function, we must have $P_n(1)=1$ for all $n \ge t$. But

$$P_n(1) \sim \alpha_1(1) + \sum_{2 \le k \le t} \alpha_k(1) \frac{\Gamma(t+1)}{\Gamma(\lambda_k+t+1)} n^{\lambda_k}$$

As $n \to \infty$, the summation goes to zero, so the limit of $\alpha_1(1)$ must be one. But $\alpha_1(1)$ does not depend on n, so it must be equal to one.

Now this implies that, for all $n \ge t$, we must have

$$\sum_{2 \le k \le t} \alpha_k(1)\, \pi_t^n(\lambda_k) = 0$$

This in turn implies that, for all $n \ge t$,

$$\sum_{2 \le k \le t} \alpha_k(1) \prod_{t<j \le n} (\lambda_k + j) = 0$$

or, using the notation $x^{\overline{n}} = x(x+1) \cdots (x+n-1)$,

$$\sum_{2 \le k \le t} \alpha_k(1) \lambda_k^{\overline{n-t}} = 0$$

We can write the first t-1 equations in matrix form as follows:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_2 & \lambda_3 & \cdots & \lambda_t \\ \lambda_2^{\overline{2}} & \lambda_3^{\overline{2}} & \cdots & \lambda_t^{\overline{2}} \\ \cdot & \cdot & \cdots & \cdot \\ \lambda_2^{\overline{t-2}} & \lambda_3^{\overline{t-2}} & \cdots & \lambda_t^{\overline{t-2}} \end{pmatrix} \begin{pmatrix} \alpha_2(1) \\ \alpha_3(1) \\ \cdot \\ \cdot \\ \alpha_t(1) \end{pmatrix} = 0$$

Let us call this matrix $\overline{V}$. Now consider the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_2 & \lambda_3 & \cdots & \lambda_t \\ \lambda_2^2 & \lambda_3^2 & \cdots & \lambda_t^2 \\ \cdot & \cdot & \cdots & \cdot \\ \lambda_2^{t-2} & \lambda_3^{t-2} & \cdots & \lambda_t^{t-2} \end{pmatrix}$$

whose determinant is $\prod_{2 \le i < j \le t} (\lambda_i - \lambda_j)$. There is a simple relation between these two matrices. If we denote

$$S = \begin{pmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & & & & \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \end{bmatrix} & & & \\ \begin{bmatrix} 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 2 \\ 1 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \end{bmatrix} & & \\ \cdot & \cdot & \cdot & \cdot & \\ \begin{bmatrix} t-2 \\ 0 \end{bmatrix} & \begin{bmatrix} t-2 \\ 1 \end{bmatrix} & \begin{bmatrix} t-2 \\ 2 \end{bmatrix} & \cdots & \begin{bmatrix} t-2 \\ t-2 \end{bmatrix} \end{pmatrix}$$

where the $\left[{n \atop k}\right]$ are the Stirling numbers of the first kind, then

$$\bar{V} = S\ V$$

But det S=1, because $\left[{n \atop n}\right]$=1 for all n. Therefore, det $\bar{V}$ = det V. Since all the eigenvalues are different, the determinant is nonzero and the system of equations has only a trivial solution, $\alpha_2(1) = \cdots = \alpha_t(1) = 0$. ■

Lemma 3.5

$C_n' \sim \lambda_1'(1)\ H_n$

$V_n' \sim (\lambda_1'(1) + \lambda_1''(1))\ H_n$

Proof

By differentiating the expression for $P_n(z)$ given in Lemma 3.3, evaluating at $z=1$, applying Lemma 3.4 and ignoring lower order terms. ■

Therefore, we have found that the average search time and its variance depend only on the first two derivatives of $\lambda_1(z)$, the eigenvalue of $H(z)$ that is zero for $z=1$.

The following lemma tells us how to find these derivatives.

Lemma 3.6

Let $\lambda(z)$ be such that $p(\lambda(z),z)=0$. Then

$$\lambda'(z) = -\frac{\frac{\partial p}{\partial z}}{\frac{\partial p}{\partial \lambda}}$$

$$\lambda''(z) = \frac{2\frac{\partial p}{\partial z}\frac{\partial p}{\partial \lambda}\frac{\partial^2 p}{\partial \lambda \partial z} - (\frac{\partial p}{\partial z})^2\frac{\partial^2 p}{\partial \lambda^2} - (\frac{\partial p}{\partial \lambda})^2\frac{\partial^2 p}{\partial z^2}}{(\frac{\partial p}{\partial \lambda})^3}$$

Proof

Since $p(\lambda,z)=0$, we have

$$dp = \frac{\partial p}{\partial \lambda} d\lambda + \frac{\partial p}{\partial z} dz = 0$$

and from this we get

$$\frac{d\lambda}{dz} = -\frac{\frac{\partial p}{\partial z}}{\frac{\partial p}{\partial \lambda}}$$

The second derivative can now be easily obtained by differentiating this expression. ■

It is now easy, using this lemma, to obtain expressions for C_n' and V_n'. We can rewrite the characteristic polinomial as

$$p(\lambda,z) = \frac{(\lambda+t+1)\cdots(\lambda+2t)}{(t+1)\cdots(2t)} - z = \pi_t^{2t}(\lambda) - z$$

From this, we get the partial derivatives

$$\frac{\partial p}{\partial z} = -1$$

$$\frac{\partial^2 p}{\partial z^2} = 0$$

$$\frac{\partial^2 p}{\partial z\, \partial \lambda} = 0$$

$$\frac{\partial p}{\partial \lambda} = \pi_t^{2t}(\lambda) \sum_{t<j\le 2t} \frac{1}{\lambda+j} \bigg|_{\substack{z=1\\ \lambda=0}} = H_{2t} - H_t$$

$$\frac{\partial^2 p}{\partial \lambda^2} = \pi_t^{2t}(\lambda) \left\{ \left(\sum_{t<j\le 2t} \frac{1}{\lambda+j} \right)^2 - \sum_{t<j\le 2t} \frac{1}{(\lambda+j)^2} \right\} \bigg|_{\substack{z=1\\ \lambda=0}}$$

$$= (H_{2t} - H_t)^2 - (H_{2t}^{(2)} - H_t^{(2)})$$

Putting all these results together, we finally have

Theorem 3.1

$$C_N' \sim \frac{1}{H_{2t} - H_t} H_N$$

$$V_N' \sim \frac{H_{2t}^{(2)} - H_t^{(2)}}{(H_{2t} - H_t)^3} H_N \qquad \blacksquare$$

Table 1 shows the value of these two coefficients for some values of t. For purposes of comparison with completely balanced binary search trees we also tabulate $\ln 2/(H_{2t} - H_t)$.

t	$\frac{1}{H_{2t}-H_t}$	$\frac{\ln 2}{H_{2t}-H_t}$	$\frac{H_{2t}^{(2)}-H_t^{(2)}}{(H_{2t}-H_t)^3}$
1	2	1.3863	2
2	1.7143	1.1883	0.8746
3	1.6216	1.1240	0.5555
4	1.5760	1.0924	0.4063
5	1.5489	1.0736	0.3201
⋮	⋮	⋮	⋮
∞	1.4427	1	0

Table 1
Coefficients for C_N' and V_N'

4. References

1. Adel'son-Vel'skii, G.M. and Landis, E.M., "An Algorithm for the Organization of Information," *Doklady Akademia Nauk SSSR* **146**, 2(1962), 263-266. English translation in *Soviet Math. Doklady* **3**(1962), 1259-1263.
2. Bell, C.J., "An Investigation into the Principles of the Classification and Analysis of Data on an Automatic Digital Computer," Doctoral Dissertation, Leeds University, 1965.
3. Guibas, L.J. and Sedgewick, R., "A Dichromatic Framework for Balanced Trees," *Proceedings of the 19th FOCS*, Ann Arbor, Michigan, **19**, 8-21, Oct. 1978.
4. Itai, A. and Rodeh, M., "Modified Binary Search Trees," Tech. Report 182, Dept. of Computer Science, Technion-IIT, Haifa, Israel, August 1980.
5. Knuth, D.E., *The Art of Computer Programming*, vol. 3 *Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.
6. Nievergelt, J. and Reingold, E.M., "Binary Search Trees of Bounded Balance," *SIAM J. Comp.* **2**, 1(1973), 33-43.
7. Sedgewick, R., "Quicksort," Ph.D. Thesis, Computer Science Dept., Stanford University, Report STAN-CS-75-492, May 1975.
8. Walker, A. and Wood, D., "Locally Balanced Binary Trees," *The Computer Journal* **19**, 4(Nov. 1967), 322-325.