Extremal Weighted Path Lengths in Random Binary Search Trees

Rafik Aguech[1] Nabil Lasmar[2] Hosam Mahmoud[3]

December 18, 2008

Abstract. We consider weighted path lengths to the extremal leaves in a random binary search tree. When linearly scaled, the weighted path length to the minimal label has Dickman's infinitely divisible distribution as a limit. By contrast, the weighted path length to the maximal label needs to be centered and scaled to converge to a standard normal variate in distribution. The exercise shows that path lengths associated with different ranks exhibit different behaviors depending on the rank. However, the majority of the ranks have a weighted path length with average behavior similar to that of the weighted path to the maximal node.

AMS subject classifications. Primary: 05C05, 60C05; secondary: 60F05, 68P05, 68P10, 68P20.

Key words. Random trees, path length, recurrence, infinitely divisible distribution, reflection principle.

1 Introduction

Various sorts of extremal path lengths in binary search trees have been studied, owing to their importance as interpretations of some analyses of algorithm (mostly in the areas of searching and sorting). For example, the height

[1] Faculté des Sciences de Monastir, Département de mathématiques, 5019 Monastir, Tunisia. E-mail: rafikaguech@ipeit.rnu.tn

[2] Institut préparatoire aux études d'ingénieurs de Tunis, Département de mathématiques, IPEIT, Rue lelnahrou-Montfleury, Tunis, Tunisia. E-mail: nabillasmar@yahoo.fr

[3] Department of Statistics, The George Washington University, Washington, D.C. 20052, U.S.A. E-mail: hosam@gwu.edu

of the binary search tree is considered in various sources for its role as a global measure of worst case search in a random tree (see Robson, 1979, Mahmoud and Pittel, 1984, Pittel 1984, Devroye, 1986–7, Drmota 2001–2, and Reed, 2003). At the other end of the spectrum, the length of the shortest root-to-leaf path is considered as a measure of optimism for the best search time (see Pittel (1984)). Many of these results are surveyed in sorting textbooks such as Knuth (1998) and Mahmoud (2000).

The above-mentioned extremal path lengths have a common thread: They all are the "raw" depth of some extremal leaf in the tree. We are concerned in this investigation with "weighted" extremal path lengths, where nodes on the path have types of contribution to the path length other than a mere count of their incoming edge, such as, for example, contributing their own value. The path lengths involved have other interpretations as quantities underlying certain algorithms.

Some algorithm may go down a path from the root of a binary search tree of size n to the node ranked j, collecting the sum of the values encountered. We investigate in this paper the distribution of such paths in some extreme cases. Let $W_j(n)$ be the value of the path length associated with traversing the tree from its root to the node labeled j, while aggregating the values on the path. We shall see that $W_1(n)$, when appropriately scaled, has Dickman's infinitely divisible distribution (a result that parallels in some way the Dickman distribution associated with finding the smallest item via th one-sided Quicksort (the so-called Quickselect algorithm)); see Mahmoud, Modarres, and Smythe (1995), and Hwang and Tsai (2002). By contrast, $W_n(n)$, when appropriately centered and scaled, converges in distribution to a normal variate. The exercise demonstrates that there is a variety of different distributions associated with $W_j(n)$ for different values of j.

2 Scope

A binary tree is a hierarchical structure of nodes each having no children, one left child, one right child, or two children (one left and one right). The nodes of such a tree can be labeled from some ordered set, say the natural numbers. The tree can further be endowed with a *search property* (to support fast searching of the items (also called *keys*) stored in it), which imposes the restriction on the labeling scheme that the label of any node is larger than the labels in its left subtree and no greater than any label in its right subtree.

For definitions and combinatorial properties see Mahmoud (1992), and for applications in sorting see Knuth (1998) or Mahmoud (2000).

A binary search tree is constructed from the permutation $(\pi_1, \ldots, \pi_n)$ of $\{1, 2, \ldots, n\}$ by the following algorithm. The first element of the permutation is inserted in an empty tree, a root node is allocated for it. A subsequent element π_j (with $j \geq 2$) is directed to the left subtree if $\pi_j < \pi_1$, otherwise it is directed to the right subtree. In whichever subtree π_j goes, it is subjected to the same insertion algorithm recursively, until it is inserted in an empty subtree, in which case a node is allocated for it and linked appropriately as a left (right) child if its rank is less than (at least as much as) the value of the last node on the path. Figure 1 illustrates the tree constructed from the random permutation $(5, 8, 7, 3, 9, 1, 6, 2, 4)$.

Several models of randomness are in common use on binary trees. The uniform model in which all trees are equally likely has been proposed for applications in formal languages, compilers, computer algebra, etc. (see Kemp (1984)). However, for the searching and sorting algorithms alluded to the *random permutation model* is considered to be more appropriate. In this model of randomness we assume that the tree is built from permutations of $\{1, \ldots, n\}$, where a uniform probability model is imposed on the *permutations* instead of the trees. When all $n!$ permutations are equally likely or *random*, binary search trees are not equally likely. Several permutations give rise to the same tree, favoring shorter and well balanced trees rather than scrawny and tall shapes, which is a desirable property in searching and sorting algorithms (see Mahmoud (1992)). The term *random tree* (and occasionally just the tree) will refer to a binary search tree built from a random permutation. The random permutation model is not really restrictive, as it covers a rather wide variety of instances, such as when the input is a sample drawn from *any* continuous probability distribution, and the construction algorithm is concerned only with the ranks of the keys, not their actual values.

We study the weighted path length leading to the rightmost and leftmost nodes. For instance, in the tree of Figure 1, $W_1(9) = 5 + 3 + 1 = 9$, $W_2(9) = 5 + 3 + 1 + 2 = 11$, $\ldots$, $W_9(9) = 5 + 8 + 9 = 22$.

The paper is organized as follows. In Section 3 we study the weighted path length leading from the root to the minimal label in the tree and show that it has Dickman's distribution (after suitable scaling). The weighted path length leading to the maximal label n is investigated separately in Section 4 where we explore a useful reflection principle. It is shown in Section 4 that the weighted path length to the maximal label converges in distribution to the

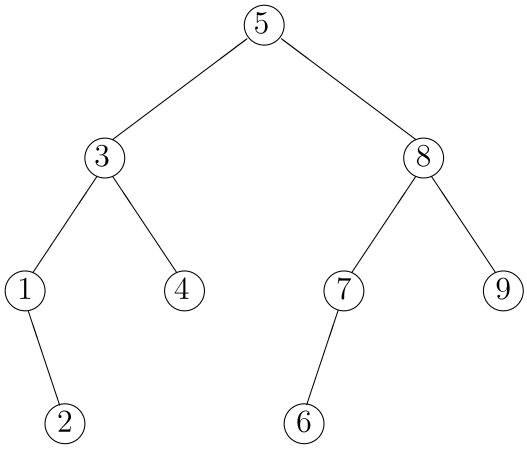

Figure 1: A binary search tree.

normal random variate (after appropriate centering and scaling). Section 5 gives some concluding remarks where a brief derivation of the average is given for the rest of the cases $1 < i < n$.

3 Weighted path to the minimal label

Let L_n be the number of items that appear in the left subtree, and thus $L_n + 1$ is the value of the root. For $n \geq 1$, the stochastic recurrence

$$W_1(n) = L_n + 1 + W_1(L_n)$$

represents the weighted path length from the root to the node labeled 1 (that is, the sum of the collection of values on the leftmost path in the tree). Let $\phi_n(t)$ be the characteristic function of $W_1(n)$. By conditioning the stochastic recurrence, we obtain

$$\begin{aligned}\phi_n(t) &= \mathbf{E}[e^{(L_n+1+W_1(L_n))it}] \\ &= \sum_{\ell=0}^{n-1} \mathbf{E}[e^{(\ell+1+W_1(\ell))it}]\, \mathbf{P}(L_n = \ell)\end{aligned}$$

$$= \frac{1}{n}\sum_{k=1}^{n} e^{kt}\,\phi_{k-1}(t),$$

valid for all $n \geq 1$. This telescoping sum is amenable to the differencing scheme—subtract a version of the last recurrence with $n-1$ from one with n to obtain

$$\phi_n(t) = \frac{n-1+e^{int}}{n}\phi_{n-1}(t),$$

which can be unwound by direct iteration to give

$$\phi_n(t) = \prod_{k=1}^{n} \frac{k-1+e^{ikt}}{k}.$$

By differentiating this latter form once and twice (then evaluating at $t=0$) we obtain the mean and the second moment.

Proposition 1 *Let $W_1(n)$ be the weighted path length from the root to the least ranked label in a binary search tree built from a random permutation. We then have*

$$\begin{aligned}\mathbf{E}[W_1(n)] &= n;\\ \mathbf{Var}[W_1(n)] &= \frac{n(n+1)}{2} \sim \frac{1}{2}n^2.\end{aligned}$$

Guided by the rate of growth of the variance, we next proceed to argue the infinite divisibility of $n^{-1}W_1(n)$. We take the natural logarithm of the characteristic function, and write it in asymptotic form (as $t \to 0$)

$$\begin{aligned}\ln(\phi_n(t)) &= \sum_{k=1}^{n} \ln\Big(1+\frac{e^{ikt}-1}{k}\Big)\\ &= \sum_{k=1}^{n}\Big[\frac{e^{ikt}-1}{k} + O\Big(\Big(\frac{e^{ikt}-1}{k}\Big)^2\Big)\Big]\\ &= O(nt^2) + \sum_{k=1}^{n}\frac{e^{ikt}-1}{k}.\end{aligned}$$

Since the rate of growth of the standard deviation is n, one expects that $W_1(n)/n$ converges to a limit in distribution. At the level of characteristic function, this means changing the scale from t to t/n. Let $v = t/n$. This entails (for fixed v)

$$\ln\Big(\phi_n\Big(\frac{v}{n}\Big)\Big) \;=\; O\Big(\frac{1}{n}\Big) + \sum_{k=1}^{n} \frac{e^{ikv/n} - 1}{k}.$$

The O term converges to 0, and the remaining sum approaches

$$\int_0^1 \frac{e^{iuv} - 1}{u}\, du.$$

We thus have the convergence

$$\phi_n\Big(\frac{v}{n}\Big) \to \exp\Big(\int_0^1 \frac{e^{iuv} - 1}{u}\, du\Big), \qquad \text{as } n \to \infty.$$

The characteristic function

$$\psi_X(v) = \exp\Big(\int_0^1 \frac{e^{iuv} - 1 - iuv}{u}\, dv\Big)$$

is that of Dickman's infinitely divisible random variable X in Kolmogorov's canonical form (see Billingsley (1995; P. 372)). That is,

$$\begin{aligned}
\phi_n\Big(\frac{v}{n}\Big) &\to e^{\int_0^1 (iv - iv)\, du} \exp\Big(\int_0^1 \frac{e^{iuv} - 1}{u}\, du\Big) \\
&= e^{iv} \exp\Big(\int_0^1 \frac{e^{iuv} - 1 - iuv}{u}\, du\Big) \\
&= \mathbf{E}[e^{i(1+X)v}].
\end{aligned}$$

We have arrived at the main result of this section.

Theorem 1 *Let $W_1(n)$ be the weighted path length from the root to the least ranked label in a binary search tree built from a random permutation. Then,*

$$\frac{W_1(n)}{n} \xrightarrow{\mathcal{D}} 1 + X,$$

where X is Dickman's random variable.

Remark: The limiting random variable for $n^{-1}W_1(n)$ bears some similarity to the limiting random variable for $n^{-1}C_n^{[1]}$, the normalized number of comparisons required by Quickselect to find the least item in a random input (of size n) with ranks following the random permutation model. It is shown in Mahmoud, Modarres and Smythe (1995) that $n^{-1}C_n^{[1]}$ converges in distribution to $2 + X$. Thus, asymptotically, the distribution of $n^{-1}C_n^{[1]}$ behaves like that of $1 + n^{-1}W_1(n)$.

4 Weighted path to the maximal label

Let us introduce a reflection operation, which may generally be useful in this type of problems. In a binary search tree T_n of size n, exchange the right and left children of every node, starting at the root and progressing recursively toward the leaves to obtain the *reflected tree* T'_n. This reflection concerns only the shape of the tree, and not the labels. One can think of this operation as if a two-sided mirror has been placed on a vertical axis passing through the root, then one sees the right subtree of T'_n as the reflection in the left side of the mirror of the left subtree of T_n, and the left subtree of T'_n as the reflection in the right side of the mirror of the right subtree of T_n. To maintain the binary search property in T'_n, we reinsert the numbers $1, \ldots, n$ in a manner consistent with the search property. For example, the reflected tree of that in Figure 1, is shown in Figure 2.

Note that, by the symmetry of binary search trees, T'_n has the same probability as T_n. That is, there are as many permutations of $\{1, 2, \ldots, n\}$ producing T_n as those producing T'_n. Observe that a key K in T_n corresponds to the value $n+1-K$ in the reflection. Let the length of the rightmost path in T_n be Q_n, and suppose the chain of values appearing on it from the root to the rightmost node (containing n) is $Y_1, Y_2, \ldots, Y_{Q_n+1}$. Observe that the rightmost path in T_n becomes a leftmost one (of the same length) in T'_n, and suppose that the corresponding labels in the reflection are $Y'_1, Y'_2, \ldots, Y'_{Q_n+1}$. This connection suggests that we can use the distribution of the path to the minimal value, which was established in Section 3, for the rightmost path as follows. We have

$$W_n(n) = \sum_{j=1}^{Q_n+1} Y_j \overset{\mathcal{L}}{=} \sum_{j=1}^{Q_n+1} (n + 1 - Y'_j) = (Q_n + 1)(n + 1) - W_1(n). \qquad (1)$$

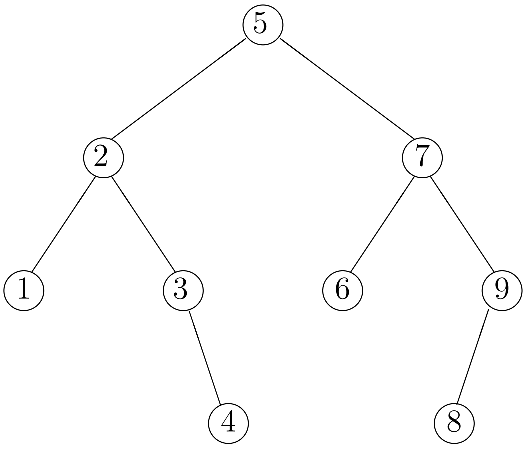

Figure 2: The reflection of the tree of Figure 1

We can now quickly develop a Gaussian law for $W_n(n)$, from known results about Q_n. The latter variable is known to be asymptotically normal, satisfying

$$\frac{Q_n - \ln n}{\sqrt{\ln n}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1);$$

see Devroye (1988). This indicates that centering and scaling the relation (1) with asymptotic mean and standard deviation of nQ_n will yield a limit distribution. Let

$$W_n^* := \frac{W_n(n) - n\ln n}{n\sqrt{\ln n}} \overset{\mathcal{L}}{=} \frac{nQ_n - n\ln n}{n\sqrt{\ln n}} + \frac{Q_n}{n\sqrt{\ln n}} + \frac{n+1}{n\sqrt{\ln n}} - \frac{W_1(n)}{n\sqrt{\ln n}}.$$

According to Theorem 1, we have

$$\frac{W_1(n)}{n\sqrt{\ln n}} \xrightarrow{a.s.} 0,$$

indicating that the main contribution in $W_n(n)$ comes from the length of the rightmost path. Also, according to the limit law of Q_n, $Q_n/(n\sqrt{\ln n}) \xrightarrow{a.s.} 0$, and of course $(n+1)/(n\sqrt{\ln n}) \xrightarrow{a.s.} 0$. Hence,

$$W_n^* \xrightarrow{\mathcal{D}} \mathcal{N}(0,1).$$

5 Conclusion

The useful notation $j \,\triangleright\, k$ will stand for the event that the label j is encountered on the path to k, thus contributing its value to the weighted path length $W_k(n)$. In view of this notation, the weighted path length of the node k is

$$W_k(n) = \sum_{j=1}^{n} j\mathbf{1}_{\{j \,\triangleright\, k\}}. \tag{2}$$

The indicators involved are dependent, and it is not easy to determine limit distributions from this representation. Even computations such as the variance are daunting. But of course, the representation being a sum, the average is no major obstacle. We develop this average, to use as a benchmark for what falls between the two extremes.

Lemma 1

$$P(j \,\triangleright\, k) = \frac{1}{|k-j|+1}.$$

Proof. Suppose $j < k$. A property of binary search trees is that when $j \,\triangleright\, k$, all the numbers in $A_{kj} = \{j, j+1, \ldots, k\}$ appear after j (See Devroye and Neininger (2004) for a discussion via the theory of records). It suffices to count B_n, the number of permutations favorable to the event $j \,\triangleright\, k$. If j appears at position p in a favorable permutation, there must be at least $k-j$ positions past p to receive the numbers in $A_{kj} - \{j\}$. Thus, $n - p \geq k - j$. To complete the construction of a favorable permutation, choose any of these $j - k$ positions for the elements of $A_{kj} - \{j\}$, and permute its $j - k$ numbers over these positions in $(k-j)!$ ways. Now permute the remaining $n-(k-j+1)$ elements in an unrestricted way over the remaining $n - (k - j + 1)$ positions. Therfore,

$$\begin{aligned} B_n &= \sum_{p=1}^{n} \binom{n-p}{k-j} (k-j)!\,(n-k+j-1)! \\ &= (k-j)!\,(n-k+j-1)! \binom{n}{k-j+1} \\ &= \frac{n!}{k-j+1}. \end{aligned}$$

The argument for $j > k$ is symmetrical, with j and k exchanging roles. $\square$

It follows from Lemma 1 and the representation (2) that

$$\mathbf{E}[W_k(n)] = \sum_{j=1}^{k-1} \frac{j}{k-j+1} + k + \sum_{j=k+1}^{n} \frac{j}{j-k+1}.$$

The substitution $m = k - j + 1$ in the first sum together with a symmetrical one in the second sum gives the result in a simple form:

$$\mathbf{E}[W_k(n)] = (k-1)H_{n-k+1} + n - 3k + 1 + (k+1)H_k,$$

where H_r is the rth harmonic number $\sum_{s=1}^{r} 1/s$.

The form for $\mathbf{E}[W_k(n)]$ is for the entire spectrum of nodes. For low-indexed nodes $k = o(n)$, we have

$$\mathbf{E}[W_k(n)] \sim n,$$

whereas for nodes with high index $k = n - o(n)$,

$$\mathbf{E}[W_k(n)] \sim n \ln n,$$

but if $k \sim \alpha n$, for $0 < \alpha < 1$, the asymptotic approximation is

$$\mathbf{E}[W_k(n)] \sim 2\alpha n \ln n,$$

which indicates that the majority of the medium range indexes lean toward the behavior of the weight of the path length to the maximal node, rather than the linear order of magnitude associated with the minimal node. This may ultimately be reflected in the average distribution across all the nodes. For instance, if we select a node randomly in the tree, its index K_n will be uniformly distributed on the set $\{1, \ldots, n\}$, and consequently its weighted path length has the average

$$\mathbf{E}[W_{K_n}(n)] = \sum_{j=1}^{n} \frac{\mathbf{E}[W_j(n)]}{n} \sim n \ln n \int_0^1 2\alpha \, d\alpha = n \ln n.$$

with an order of magnitude just like that of the weight of the path length to the maximal node

References

[1] Billingsley, P. (1995). *Probability and Measure*. Wiley, New York.

[2] Devroye, L. (1986). A note on the height of binary search trees. *Journal of the ACM*, **33**, 489–498.

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

[4] Devroye, L. (1988). Applications of the theory of records in the study of random trees. *Acta Informatica*, **26**, 123–130.

[5] Devroye, L. and Neininger, R. (2004). Distances and finger search in random binary search trees. *SIAM Journal on Computing*, **33**, 647–658.

[6] Drmota. M. (2001). An analytic approach to the height of binary search trees. *Algorithmica* **29**, 89–119.

[7] Drmota. M. (2002). The variance of the height of binary search trees. *Theoretical Computer Science*, **270**, 913–919.

[8] Hwang, H. and Tsai, T. (2002). Quickselect and Dickman function. *Combinatorics, Probability and Computing*, **11**, 353–371.

[9] Kemp, R. (1984). *Fundamentals of the Average Case Analysis of Particular Algorithms*. Wiley-Teubner Series in Computer Science, John Wiley & Sons, New York.

[10] Knuth, D. (1998). *The Art of Computer Programming*, Vol. 3: *Sorting and Searching*, 2nd ed. Addison-Wesley, Reading, Massachusetts.

[11] Mahmoud, H. (1992). *Evolution of Random Search Trees*. Wiley, New York.

[12] Mahmoud, H. (2000). *Sorting: A Distribution Theory*. Wiley, New York.

[13] Mahmoud, H., Modarres, R. and Smythe, R. (1995). Analysis of quickselect: An algorithm for order statistics. *RAIRO: Theoretical Informatics and Its Applications*, **29**, 255–276.

[14] Mahmoud, H. and Pittel, B. (1984). On the most probable shape of a search tree grown from a random permutation. *SIAM Journal on Algebraic and Discrete Methods*, **5**, 69–81.

[15] Pittel, B. (1984). On growing random binary trees. *Journal of Mathematical Analysis and its Applications*, **103**, 461–480.

[16] Reed, B. (2003). The height of a random binary search tree. *Journal of the Association for Computing Machinery*, **50**, 306–332.

[17] Robson, J. (1979). The height of binary search trees. *The Australian Computer Journal*, **11**, 151–153.