The Annals of Applied Probability
1998, Vol. 8, No. 1, 112–130

NORMALITY OF TREE-GROWING SEARCH STRATEGIES

BY RUSSELL LYONS[1] AND KEVIN ZUMBRUN[2]

Indiana University

We study the class of tree-growing search strategies introduced by Lent and Mahmoud, searches for which data are stored in a *deterministic* sequence of tree structures (e.g., linear search in forward order). Specifically, we study the conditions under which the number of comparisons needed to sort a sequence of randomly ordered numbers is asymptotically normal. Our main result is a sufficient condition for normality in terms of the growth rate of tree height alone; this condition is easily computed and is satisfied by all standard deterministic search strategies. We also give some examples of normal search strategies with surprisingly small variance, in particular, much smaller than is possible for the class of consistent strategies that are the focus of the work by Lent and Mahmoud.

1. Introduction. The problem we consider arises from the study of computer search–sort algorithms. A question of theoretical interest is the number of comparisons C_n needed to sort the first n elements of a linear stream of data consisting of random real numbers. The expectation and variance of C_n give a partial description; if it can be shown that C_n is asymptotically normal, then these two parameters (together, perhaps, with the rate of convergence) provide a good description of the asymptotic behavior of C_n.

For a class of algorithms they called "tree-growing," Lent and Mahmoud [2] showed that this question reduces to a combinatorial problem involving sequences of trees and identified a subclass of "consistent," or "self-similar," tree-growing search strategies that do have asymptotically normal behavior. Here, we study this problem without the simplifying assumption of consistency. We give a general and easy-to-compute condition sufficient for asymptotic normality that covers perhaps all practical deterministic search strategies. In particular, we recover the results of [2] as special cases and resolve certain issues they left open. In the process, we show that the behavior of consistent searches is quite special among general tree-growing searches.

We first describe the reduction to the combinatorial problem. In the following, trees are assumed to be extended (or full) binary, which means that when the tree is oriented so that edges lead away from a distinguished vertex called the root, each node has either two or no edges leading out of it. The nodes to which these edges lead are called the left and right children of the parent node from which they emanate. The leaves, that is, those nodes with no edges

Received August 1996; revised May 1997.
[1]Supported in part by NSF Grant DMS-93-06954.
[2]Supported in part by NSF Grant DMS-91-07990 and ONR Grant N00014-94-1-0456.
AMS 1991 *subject classifications*. Primary 68P10; secondary 60F05.
Key words and phrases. Sort, insert, limit distribution.

leading out, are called external nodes; the others are internal nodes. It is well known and easily verified by induction that the number of external nodes is one more than the number of internal nodes. Contrary to the usual convention in computer science (in the U.S.), we refer to our trees as growing upwards from the root. The level of a node denotes its distance from the root and the height of a tree denotes its maximal (external) node level. Thus, the level of the root is 0.

A (tree-growing) search strategy is specified by a *deterministic* sequence of trees $\langle T_n \rangle$ with the properties that T_n has n external nodes and is obtained from T_{n-1} by converting one external node to an internal node and a pair of external nodes. At each stage in the sorting process, we assume that the $n-1$ data elements already sorted have been stored in the internal nodes of T_n in the unique configuration for which the data value of each left child is less than the value of its parent, and the data value of each right child is greater than the value of its parent. When the nth datum arrives, its rank relative to the first $n-1$ data elements can then be found by starting at the root and moving always to the left or right child according as this datum is less than or greater than the value at the current location. When one reaches an external node (where no datum is stored), the new datum is *temporarily* stored there, recording its precise rank among the first n data elements. Unless the external node where the nth datum is stored happens to be the one converted to an internal node in T_{n+1}, the data is then reconfigured into the next prescribed structure T_{n+1}, and the process is continued. For example, linear search in forward order stores the data in linear order, no matter in what order they arrive. See Figure 1 for another example and [2] or Section 2 for analysis of other examples.

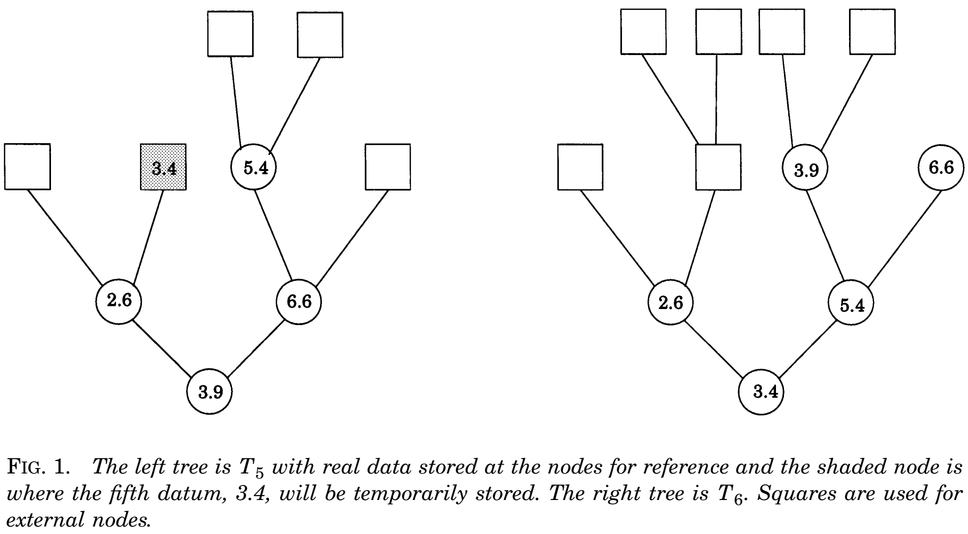

FIG. 1. *The left tree is T_5 with real data stored at the nodes for reference and the shaded node is where the fifth datum, 3.4, will be temporarily stored. The right tree is T_6. Squares are used for external nodes.*

We emphasize the distinction between the deterministic search tree strategies studied here and the more widely studied class of *random* search tree strategies, described, for example, in [3]. For a given n, a random search strategy allows for data storage in any of a family of usually "balanced" (hence efficient) tree structures. This is accomplished most of the time simply by placing the newest datum in an external node of T_{n-1}; occasionally, the tree must be rebalanced to stay within the class of admissible data structures. By contrast, the deterministic strategies prescribe ahead of time only one possible tree structure for each value of n, which is not necessarily well balanced. Indeed, the concept of tree-growing search strategy was introduced in [2] precisely to study the situation in which the data structure is dictated by considerations other than maximum efficiency, such as ease in progamming or the nature of the storage device (the example is given in [2] of a linear tape drive). Further discussion of applications and motivation for the problem considered may be found in [2].

Besides possible imbalance, (deterministic) tree-growing searches incur the penalty that the data may need to be completely reconfigured for each n instead of merely rebalanced, at a cost of $O(n)$ pointer changes. Depending on the application, that is, the relative cost of data movement and comparison operations, this may dominate the cost of the search; on the other hand, the cost of comparisons is typically greater than that of data movement, by several orders of magnitude in some cases. Following [2], we ignore the cost of reconfiguration and *analyze only the number of data comparisons*. This can be interpreted as restriction to the high comparison-cost regime and large but not infinite n.

Now, let X_n be the number of comparisons needed to enter the nth data element, that is, the level of the external node of T_n at which this datum is temporarily stored (before reconfiguration). Thus, the total number of comparisons made in sorting the first n data is the partial sum $C_n := \sum_{i=1}^{n} X_i$. If the data form a stream of i.i.d. continuously distributed numbers, then the rank of the nth datum among the first n is uniformly distributed among the integers $1, \ldots, n$, and is independent of the relative ranks of previous elements. Thus, the level X_n of the nth datum is the level of a uniformly chosen external node of T_n, and, because $\langle T_n \rangle$ are deterministic, the random variables X_n are independent for different n.

This reduces our problem to a purely combinatorial question: *let $\langle T_n \rangle$ be a sequence of trees, ordered as described above. Define independent random variables X_n to be the levels of uniformly chosen external nodes of T_n, and denote their partial sums as $C_n := \sum_{i=1}^{n} X_i$. Is C_n asymptotically normal in the sense that $(C_n - \mathbf{E}[C_n])/\sqrt{\mathrm{Var}(C_n)}$ converges in distribution to the standard normal?*

Hereafter, we restrict our discussion to this combinatorial problem, referring the reader to [2] for further background. We note, however, that if we replace external nodes at level k by balls in an urn numbered k, we obtain an amusing formulation of the problem that involves no trees: suppose that there are urns numbered by the nonnegative integers. We begin with one ball in urn number 0. There is some deterministic scheme described by a function $f(n)$ that at each time $n \geq 1$, takes a ball from urn number $f(n)$ and replaces it with two balls in urn number $f(n)+1$. Let X_n be the urn number of a randomly (that is, independently and uniformly) chosen ball at time n and $C_n := \sum_{i=1}^{n} X_i$. Which functions f lead to asymptotically normal behavior of C_n?

NOTATION. Let h_n be the height of T_n and μ_n be the mean of X_n, that is, the mean level of the external nodes of T_n. Define the tree variance σ_n^2 to be the variance of X_n and the procedural variance s_n^2 to be the variance of C_n, or

$$s_n^2 = \sum_{i=1}^{n} \sigma_i^2.$$

We also define the growth function $n(h) := \max\{n;\ h_n = h\}$ and its first difference, the growth rate $m(h) := \operatorname{card}\{n;\ h_n = h\}$. (In [2], these were denoted U_h and m_h, respectively.) We shall write $f(n) = \Omega(g(n))$ or $f(n) \gg g(n)$ to mean that $|f(n)/g(n)|$ is bounded below by a positive number and $f(n) = O(g(n))$ or $f(n) \ll g(n)$ to mean that $|f(n)/g(n)|$ is bounded above. We shall write $f(n) \asymp g(n)$ to mean that $f(n)/g(n)$ is bounded above and below by two positive numbers. All logarithms are to the base 2. For any real number x, we denote by $\lfloor x \rfloor$ and $\lceil x \rceil$ the consecutive integers satisfying $\lfloor x \rfloor \leq x < \lceil x \rceil$.

A necessary and sufficient condition for asymptotic normality of a general sum of independent random variables is given by the Lindeberg–Feller theorem:

PROPOSITION 1.1 (Lindeberg–Feller theorem). *Let X_n $(n \geq 1)$ be independent random variables with finite second moments. Let $\sigma_n^2 := \operatorname{Var}(X_n)$ and*

$$s_n^2 := \sum_{i=1}^{n} \sigma_i^2.$$

For $\varepsilon > 0$, set

$$\tilde{s}_{n,\varepsilon}^2 := \sum_{i=1}^{n} \mathbf{E}\big[(X_i - \mathbf{E}[X_i])^2;\ |X_i - \mathbf{E}[X_i]| > \varepsilon s_n\big].$$

Assume that $\lim_{n\to\infty} s_n = \infty$ and $\lim_{n\to\infty} \sigma_n/s_n = 0$. Then $\sum_{i=1}^{n}(X_i - \mathbf{E}[X_i])/s_n$ converges in distribution to the standard normal if and only if $\forall\ \varepsilon > 0$ $\lim_{n\to\infty} \tilde{s}_{n,\varepsilon}/s_n = 0$.

See, for example, [1], Section XV.6. In the context of tree-growing searches, it is easy to see that the conditions $\lim_{n\to\infty} s_n = \infty$ and $\lim_{n\to\infty} \sigma_n/s_n = 0$ always hold. In this context, Lent and Mahmoud [2] noted the following sufficient condition for normality, repeated here with their proof. We have added a bound on the rate of convergence, defined to be the supremum of the difference between the cumulative distribution function of $\sum_{i=1}^{n}(X_i - \mathbf{E}[X_i])/s_n$ and that of the standard normal.

COROLLARY 1.2. *The condition*

$$h_n = o(s_n) \tag{1.1}$$

implies asymptotic normality of the search strategy $\langle T_n \rangle$. *Furthermore, the rate of convergence is bounded by* $6h_n/s_n$.

PROOF. The interval $[0, h_n]$ includes the levels of all the external nodes and hence their mean. Therefore, given $\varepsilon > 0$, when $h_n < \varepsilon s_n$, there are no nodes whose level differs from μ_n by more than εs_n, whence for such n, the corresponding $\tilde{s}_{n,\varepsilon} = 0$. The bound on the rate of convergence follows from Esséen's theorem ([1], Section XVI.5) since the third moment of any random variable is bounded by the product of its supremum times its second moment. □

The above corollary states that a search is normal provided that its procedural variance grows sufficiently fast relative to its height. It is basic to the analysis of [2] and to ours as well. However, note that it can only be applied after one first estimates s_n, which may be a nontrivial task.

The class of consistent tree-growing searches is defined in [2] as follows.

DEFINITION. For a given tree T, let T^L and T^R denote the descendant subtrees of the left and right children of the root of T, respectively, and let $|T|$ denote the number of external nodes in T. A tree-growing search strategy $\langle T_n \rangle$ is consistent if T_n^L is isomorphic to $T_{|T_n^L|}$ and T_n^R is isomorphic to $T_{|T_n^R|}$ for all n.

The main result of [2] is the following.

PROPOSITION 1.3. *Let* $\langle T_n \rangle$ *be a consistent strategy satisfying*

$$\lim_{n\to\infty} |T_n^L|/n \quad \textit{exists}. \tag{1.2}$$

Then $\langle T_n \rangle$ *satisfies* (1.1), *whence is normal*.

Lent and Mahmoud implicitly conjectured, and proved in special cases, that the technical condition (1.2) is unnecessary. In addition, they noted that most "reasonable" tree-growing searches are normal, even if not consistent; for example, they showed by separate arguments that alternating linear and some types of Fibonacci search are normal, despite not being consistent. They conjectured, in particular, that every tree-growing Fibonaccian search is asymptotically normal.

Here, we derive a condition for normality analogous to (1.1) but involving only the growth function $n(h)$, generally an easily computed function. Define

$$D(h) := \log^4 n(h) + \sum_{k=1}^{h} k^3 \log \frac{n(k)}{n(k-1)}.$$

Our main result is the following.

PROPOSITION 1.4. $s_n^2 \geq 2^{-63} D(h_n - 1) \geq 2^{-63} \sum_{k=1}^{h_n - 1} k^3 m(k)/n(k)$.

REMARK. In Proposition 1.4, $h_n - 1$ can be replaced by h_n when $n = n(h_n)$.

An immediate consequence is the following theorem.

THEOREM 1.5. *The condition*

$$D(h)/h^2 \to \infty \tag{1.3}$$

as $h \to \infty$ *implies* (1.1) *and hence asymptotic normality. The rate of convergence to normality is bounded by* $2^{35} h_n / \sqrt{D(h_n - 1)}$.

Of course, a better lower bound for s_n leads to a better upper bound on the rate of convergence, as would a better upper bound on the maximum of $|X_i - \mathbf{E}[X_i]|$. We shall ignore rates of convergence in the sequel.

Call a strategy monotone if $\langle m(k) \rangle$ is nondecreasing.

COROLLARY 1.6. *Monotone tree-growing search strategies are asymptotically normal. More generally, searches satisfying the regularity condition*

$$\lim_{k \to \infty} k \frac{n(k) - n(\lfloor \lambda k \rfloor)}{n(k)} = \infty \tag{1.4}$$

for some $\lambda \in (0, 1)$ *are asymptotically normal.*

PROOF. Both claims follow from Theorem 1.5 and the second inequality of Proposition 1.4. For the first claim, we substitute $km(k)$ for $n(k)$ using the relation $n(k) \leq km(k)$. For the second claim, we use

$$\sum_{k=1}^{h} \frac{k^3 m(k)}{n(k)} \geq \frac{(\lambda h)^3}{n(h)} \sum_{\lambda h < k \leq h} m(k) = (\lambda h)^3 \frac{n(h) - n(\lfloor \lambda h \rfloor)}{n(h)}.$$

[Alternatively, one can show that monotonicity implies (1.4).] □

REMARK. In [2], it was proved that all consistent strategies are monotone. Thus, monotonicity generalizes the notion of consistency, and we recover Proposition 1.3 from Corollary 1.6 without using the technical hypothesis (1.2). In fact, virtually all strategies discussed in [2] are monotone, hence normal. In particular, every Fibonacci search is asymptotically normal. This verifies the conjectures of [2]. At the end of [2], the authors remark that they can prove normality when three certain conditions, unrelated to consistency, are satisfied. One of these conditions is that $n(k)/m(k) = O(k^{1+\delta})$ for some $\delta \in (0, 1)$. By virtue of Theorem 1.5 and Proposition 1.4, we see that this condition alone suffices for normality.

If the quotient in (1.4) is assumed merely to be bounded below, then normality may not hold (see Example 5.2). On the other hand, in such a case, (1.1) is then a necessary and sufficient condition for normality; a second use of Proposition 1.4 will be to show the following.

PROPOSITION 1.7. *For strategies satisfying*

$$h\frac{n(h) - n(\lfloor \lambda h \rfloor)}{n(h)} = \Omega(1) \tag{1.5}$$

for some $\lambda \in (0, 1)$, *condition* (1.1) *is equivalent to normality.*

In Section 5, we show that, although (1.3) is not equivalent to normality in the usual sense, it *is* equivalent in the sense that if (1.3) is violated, then the growth rate $m(h)$ is consistent with a nonnormal strategy $\langle T_n \rangle$. We must be content with this weaker statement since the growth rate is usually far from a complete specification of a search strategy; in fact, a given growth rate may be arise from both normal and nonnormal strategies (cf. Example 5.2).

In the final section, we show that (1.1) is not equivalent to normality in the general case. Indeed, we demonstrate the rather counterintuitive fact that there exist normal tree-growing search strategies with $s_n = O(h_n^\lambda)$ and $s_n = O(n^{\lambda/6})$ with $\lambda > 0$ as small as desired. This is quite different from the case of consistent search strategies, for which $s_n = \Omega(n^{1/2})$ [2] [and so $s_n = \Omega(h_n^{1/2})$], and such possibilities must be taken into account in our analysis in Sections 3 and 4.

2. Three examples. We begin with some simple examples, both to motivate the analysis that follows and to emphasize the distinction between consistent and general tree-growing searches.

The most basic example is linear search, which is trivially analyzed.

EXAMPLE 2.1. If $m(k) = 1$ for all k, then $s_n^2 \asymp n^3 = (h_n + 1)^3$.

Note that such trees arise from various implementations of linear search, including linear search in forward, reverse, or alternating order through the data. Different implementations give trees that differ only in the assignment of left and right to the internal children.

Another basic example is the usual binary search.

EXAMPLE 2.2. If $m(k) = 2^k$, then $s_n^2 \asymp n \asymp 2^{h_n}$.

Both these strategies are consistent (more precisely, admit consistent realizations) and fill in the trees from bottom to top, also called breadth first. (Recall that our trees grow upwards.) It is tempting to believe that filling in breadth first minimizes s_n, given a desired terminal tree T_n. In fact, one might conjecture that the greedy algorithm (which always converts the node of T_n that adds the least variance) is optimal. However, these conjectures are far from correct. Indeed, an important example for us is an unusual way to grow complete binary trees that gives much less variance than the standard breadth-first fill-in.

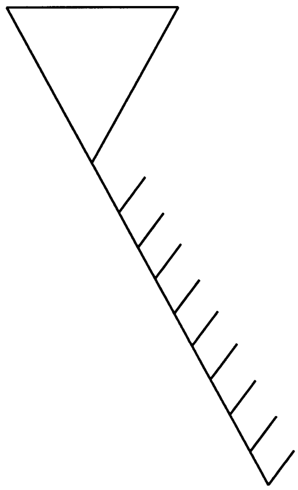

FIG. 2. *An intermediate stage in filling in a tree.*

EXAMPLE 2.3. Given $n = 2^h$, for $i \leq n$, form T_i from T_{i-1} by converting to an internal node the leftmost external node at level at most h (see Figure 2). Then $s_n^2 = O(\log^4 n) = O(h^4)$.

As opposed to the usual breadth-first method, this left-to-right, or depth-first, method does not work simultaneously for all h. However, it can be essentially repeated for a sequence of heights whose differences increase to infinity. Thus, one can arrange that

$$\liminf_{n\to\infty} s_n^2 / \log^4 n < \infty.$$

(However, the corresponding inequality

$$\limsup_{n\to\infty} s_n^2 / \log^4 n < \infty$$

cannot hold, as shown by the proof of Lemma 3.2.)

PROOF. Induction shows that for each $i \leq n$, the tree T_i has at most one right external node at each level $< h$ and has at most one left external node altogether, except for those external nodes at level h. Therefore, σ_i^2 is bounded above by the mean squared distance of the external nodes from level h, which is $O(h^3/i)$. Summing over i gives the result. □

Example 2.2 played an important role in [2]; it shows the sharpness of the bound $s_n^2 = \Omega(n)$ among consistent searches. For us, Example 2.3 will play a similarly important role. For instance, a key step in our analysis will be to establish that $s_n^2 = \Omega(\log^4 n)$ for general tree-growing search strategies; indeed, our analysis in the next section is primarily motivated by this worst-case scenario. The difference between Examples 2.2 and 2.3 indicates the strong distinction in behavior of consistent and general tree-growing searches.

3. Variance estimates. In this section, we determine lower bounds for the variance s_n^2 in terms of the growth function $n(h)$. These all arise out of the following dichotomy, which asserts that, in an approximate sense, a tree T_n must either be full binary or else contain a linear search of full height h_n.

LEMMA 3.1. *For any tree T_n, either $n \geq 2^{h_n/16}$ or else T_n contains at least $\lfloor h_n/16 \rfloor$ external nodes with level in the interval $((l-1)h_n/3, lh_n/3]$, for each $l = 1, 2, 3$, in which case*

$$\sigma_n^2 \geq \frac{h_n^3}{2^{10} n}.$$

PROOF. The assertion follows trivially if $h_n < 16$ or $n \geq 2^{h_n/16}$. Suppose instead that $h_n \geq 16$ and $h_n/\log n > 16$, so $n \geq 16$. In this case, we claim that if $0 \leq \alpha \leq h_n - 3\log n$, then there are more than $(3/2)\log n$ external nodes with level in the interval $(\alpha, \alpha + 3\log n]$, for there is a path of more than $(3/2)\log n$ internal nodes contained between levels α and $\alpha + 2\log n$. From each of these internal nodes, there branches a distinct subtree in the direction transverse to the path. Now none of these subtrees can be complete for $\lceil \log n \rceil$ levels from its root since it would then contain more than or equal to $2^{\log n} = n$ external nodes, which is the total number of external nodes in T. Hence each of these subtrees contains an external node of level less than or equal to $\log n$ from its root, whence in the interval $(\alpha, \alpha + 3\log n]$. Since there are more than $(3/2)\log n$ such subtrees, the claim is established.

Among the n external nodes of T_n, there are thus at least

$$(3/2)\log n \left\lfloor \frac{h_n/3}{3\log n} \right\rfloor \geq \lfloor h_n/16 \rfloor$$

in each of the intervals $(0, h_n/3]$, $(h_n/3, 2h_n/3]$, $(2h_n/3, h_n]$, as claimed. Since $h_n \geq 16$, we have $\lfloor h_n/16 \rfloor \geq h_n/32$. Thus, there exists a subset of the external nodes containing at least $h_n/32$ nodes from each of $(0, h_n/3]$ and $(2h_n/3, h_n]$. This subset must have variance at least $(h_n/6)^2$, and therefore T_n has variance σ_n^2 at least

$$\left(\frac{h_n}{6}\right)^2 \frac{h_n/16}{n} \geq \frac{h_n^3}{2^{10} n},$$

which completes the proof. □

LEMMA 3.2. $s_n^2 \geq 2^{-50} \log^4 n$.

PROOF. It suffices to deal with the case that $s_n^2 < 2^{-6} \log^4 n$ and $n \geq 2^{2^{12}}$ since for any $n \geq 2$, we have $s_n^2 \geq 1/4$. Note that then

$$\log(n/2) \geq 32 \log(4 \log^4 n). \tag{3.1}$$

In particular, $n \geq 4 \log^4 n$.

Our first observation is that, for $i \geq 4\log^4 n$, more than 3/5 of all external nodes of T_i must be at the same level. Otherwise, for the previous $\lfloor i/5 \rfloor$ steps, there would be at most $4i/5$ external nodes on any given level, in particular at the median level. It follows that there would be two subsets, each of $\lfloor i/10 \rfloor$ nodes, separated by at least one level. The union of these two subsets would have variance at least $(1/2)^2$. This would give a tree variance for each $4i/5 < j \leq i$ of

$$\sigma_j^2 \geq \left(\frac{1}{2}\right)^2 \frac{2\lfloor i/10 \rfloor}{j} \geq 1/40,$$

and therefore a procedural variance of at least

$$s_i^2 \geq i/200 \geq 2^{-6}\log^4 n,$$

which contradicts the original assumption.

It follows that all T_i such that $i \geq 4\log^4 n$ must have more than 3/5 of their nodes at a fixed level h_*, where h_* is independent of i. In particular, since $n \geq 4\log^4 n$, the tree T_n has at least $n/2$ nodes at level h_*, so that $2^{h_*} \geq n/2$ and therefore

$$h_* \geq \log(n/2) \geq (1/2)\log n. \tag{3.2}$$

Now, appealing to Lemma 3.1, we find that for all i between $4\log^4 n$ and $2^{h_*/16}$, we have $\sigma_i^2 \geq h_*/2^{10} i$. Summing over all such i, we get

$$\begin{aligned} s_n^2 &\geq \sum_{4\log^4 n}^{2^{h_*/16}} \frac{h_*^3}{(2^{10} i)} \geq \frac{h_*^3}{2^{10}\log e}\left(\frac{h_*}{16} - \log(4\log^4 n)\right) \\ &\geq \frac{h_*^3}{2^{11}}\left(\frac{\log(n/2)}{16} - \log(4\log^4 n)\right) \geq \frac{h_*^3}{2^{11}}\frac{\log(n/2)}{32} \\ &\geq 2^{-20}\log^4 n, \end{aligned}$$

where the last three inequalities follow from (3.2), (3.1) and (3.2). □

LEMMA 3.3. $s_n^2 \geq 2^{-62}\sum_{k=1}^{h_n-1} k^3 \log(n(k)/n(k-1))$.

PROOF. By Lemma 3.2, we have $s_n^2 \geq 2^{-50}\log^4 n$ for all n. Set N to be the largest integer less than or equal to n, such that $h_N/16\log N < 1$ if there is one, else $N := 0$. Note that if $0 < N < n$, then $N = n(h_N)$. Therefore, when $0 < N < n$, we have

$$s_N^2 \geq 2^{-50}\log^4 N \geq 2^{-62} h_N^3 \log N = 2^{-62} h_N^3 \sum_{k=1}^{h_N} \log\frac{n(k)}{n(k-1)}.$$

Combining this with the result of Lemma 3.1, we have, in case $N < n$,

$$
\begin{aligned}
s_n^2 &= s_N^2 + \sum_{i=N+1}^{n} \sigma_i^2 \\
&\geq 2^{-62} \sum_{k=1}^{h_N} k^3 \log \frac{n(k)}{n(k-1)} + 2^{-61} \sum_{k=h_N+1}^{h_n} k^3 \sum_{i=n(k-1)}^{n(k)} 1/i \\
&\geq 2^{-62} \sum_{k=1}^{h_n} k^3 \log \frac{n(k)}{n(k-1)}.
\end{aligned}
$$

In case $N = n$, we have instead

$$
\begin{aligned}
s_n^2 &\geq 2^{-50} \log^4 n \geq 2^{-62} h_n^3 \log n \geq 2^{-62} h_n^3 \sum_{k=1}^{h_n-1} \log \frac{n(k)}{n(k-1)} \\
&\geq 2^{-62} \sum_{k=1}^{h_n-1} k^3 \log \frac{n(k)}{n(k-1)}.
\end{aligned}
$$

These cases together yield the result. □

4. Proofs of the main results.

We now prove our main results.

PROOF OF PROPOSITION 1.4. The first inequality follows immediately from the estimates Lemma 3.2 and Lemma 3.3. The second inequality follows from

$$
\log \frac{n(k)}{n(k-1)} = \log\left(1 + \frac{m(k)}{n(k-1)}\right) \geq \log\left(1 + \frac{m(k)}{n(k)}\right) \geq m(k)/n(k),
$$

where the last inequality holds because $0 < m(k)/n(k) \leq 1$ (recall that logarithms are to the base 2). □

To prove Proposition 1.7, we first establish the following lemma.

LEMMA 4.1. *Let $0 < \lambda < 1$ be fixed. Set $\hat{\sigma}_n^2 := \mathrm{Var}(X_n;\ |X_n - \mu_n| \geq \lambda h_n/10)$ and $\hat{s}_n^2 := \sum_{j=1}^{n} \hat{\sigma}_j^2$. If for some h, we have $D(h) \leq (27/625)\lambda^4 h^3$, then*

$$
\hat{s}_{n(h)}^2 \geq 10^{-3} \lambda^3 h^3 \left(\frac{n(h) - n(\lfloor \lambda h \rfloor)}{n(h)}\right).
$$

PROOF. We have

$$
\begin{aligned}
D(h) &\geq \sum_{0.6\lambda h < k \leq h} k^3 \frac{m(k)}{n(k)} \geq \frac{(0.6\lambda h)^3}{n(h)} \sum_{0.6\lambda h < k \leq h} m(k) \\
&= \frac{(0.6\lambda)^3 h^3}{n(h)} \big(n(h) - n(\lfloor 0.6\lambda h \rfloor)\big),
\end{aligned}
$$

so that

$$\frac{n(h) - n(\lfloor 0.6\lambda h \rfloor)}{n(h)} \le \frac{D(h)}{(0.6\lambda)^3 h^3} \le 0.2\lambda.$$

Thus, for $0.6\lambda h \le h_j \le h$, the mean μ_j satisfies

$$\mu_j \le \lfloor 0.6\lambda h \rfloor \frac{n(\lfloor 0.6\lambda h \rfloor)}{n(h)} + h\frac{n(h) - n(\lfloor 0.6\lambda h \rfloor)}{n(h)} \le 0.6\lambda h + 0.2\lambda h = 0.8\lambda h.$$

It follows that for $\lambda h \le h_j \le h$, there are at least $h_j - \lambda h \ge 0.1\lambda h$ external nodes of T_j at level greater than or equal to $0.9\lambda h \ge \mu_j + 0.1\lambda h$. This gives a tail tree variance of

$$\hat{\sigma}_j^2 \ge (0.1\lambda h)^2 \frac{0.1\lambda h}{j} \ge \frac{10^{-3}(\lambda h)^3}{n(h)}.$$

Summing $\hat{\sigma}_j^2$ over those j with h_j between λh and h, we obtain the desired inequality. □

PROOF OF PROPOSITION 1.7. Assume that (1.5) holds for some λ. We need only show (1.1) to be necessary. Suppose that (1.1) is violated; that is, there is a constant C and a sequence $\langle H_j \rangle$ such that $s_{n(H_j)} \le CH_j$. By Proposition 1.4, we have $D(H_j) \le 2^{63} s^2_{n(H_j)} \le 2^{63} C^2 H_j^2$. For large enough j, it follows that $D(H_j) \le (27/625)\lambda^4 H_j^3$. Choose $\varepsilon < \lambda/10$. By Lemma 4.1, we have that the tail variance $\tilde{s}^2_{n(H_j),\,\varepsilon}$ in the Lindeberg–Feller theorem satisfies

$$\tilde{s}^2_{n(H_j),\,\varepsilon} \ge \hat{s}^2_{n(H_j)} \ge 10^{-3}(\lambda H_j)^3 \left(\frac{n(H_j) - n(\lfloor \lambda H_j \rfloor)}{n(H_j)} \right) = \Omega(H_j^2) = \Omega\big(s^2_{n(H_j)}\big)$$

and the strategy is therefore not asymptotically normal. □

5. Sharpness of the growth rate condition. For strategies satisfying the regularity condition (1.5), we have shown that (1.1) is a sharp condition for normality. We now show that, with an appropriate interpretation, condition (1.3) is also sharp.

PROPOSITION 5.1. *Let* $m(k) \le 2^k$ *satisfy the regularity condition* (1.5). *If* $\langle m(k) \rangle$ *violates* (1.3), *then there exists a nonnormal strategy* $\langle T_n \rangle$ *with growth rate* $\langle m(k) \rangle$.

PROOF. We shall show, more generally, that for *any* $m(k) \le 2^k$ violating (1.3), there exists a strategy $\langle T_n \rangle$ with growth rate $\langle m(k) \rangle$ violating (1.1). In the case that $\langle m(k) \rangle$ satisfies (1.5), Proposition 1.7 gives the nonnormality of $\langle T_n \rangle$.

If $\langle m(k) \rangle$ violates (1.3), then $D(H_p) \le CH_p^2$ for some constant C and some sequence of heights H_p satisfying

$$\log n(H_p) \ge n(H_{p-1}). \tag{5.1}$$

Let $\langle T_n \rangle$ be the following depth-first strategy.

For i of the form $n(k)+1$, convert the leftmost external node of T_{i-1} at the kth level to an internal node and two external nodes at level $k+1$. For all other i, if p is such that $n(H_{p-1}) < i \le n(H_p)$, then convert the leftmost external node of T_{i-1} at level $\le \lceil \log n(H_p) \rceil$. That is, subject to the growth rate $m(k)$, perform as nearly as possible the depth-first binary search of Example 2.3.

For this strategy, we claim that

$$s^2_{n(H_p)} = O(D(H_p)) = O(H_p^2),$$

which violates (1.1).

For, at each step $i \le n(H_p)$, the external nodes of T_i are the union of at most $n(H_{p-1})$ nodes at level less than or equal to $\lceil \log n(H_{p-1}) \rceil$, at most i nodes at level $\lceil \log n(H_p) \rceil$, and (as in Example 2.3) at most two nodes at every other level less than or equal to h_i. Bounding variance by mean squared distance from $\lceil \log n(H_p) \rceil$, we have therefore

$$\sigma_i^2 \le \lceil \log n(H_p) \rceil^2 \left(\frac{n(H_{p-1})}{i} \right) + h_i^2 \left(\frac{2h_i}{i} \right).$$

Summing over i, we obtain

$$\begin{aligned} s^2_{n(H_p)} &\ll \sum_{i=1}^{n(H_p)} \log^2 n(H_p) \frac{n(H_{p-1})}{i + h_i^3/i} \\ &= \sum_{k=1}^{H_p} \bigl(\log^2 n(H_p) n(H_{p-1}) + k^3 \bigr) \sum_{n(k-1)<i\le n(k)} 1/i \\ &\asymp \sum_{k=1}^{H_p} \bigl(\log^2 n(H_p) n(H_{p-1}) + k^3 \bigr) \log \frac{n(k)}{n(k-1)} \\ &= \log^3 n(H_p) n(H_{p-1}) + \sum_{k=1}^{H_p} k^3 \log \frac{n(k)}{n(k-1)} \\ &\le D(H_p), \end{aligned}$$

where the final inequality follows from (5.1). This proves the claim and we are done. □

The following example shows that normality is not completely determined by growth rate, even for strategies satisfying the regularity condition (1.5). This explains the formulation of Proposition 5.1; at the same time, it shows that there exist regular strategies that are nonnormal.

EXAMPLE 5.2. There exists a sequence $\langle m(k) \rangle$ satisfying the regularity condition (1.5), which is the growth rate of both normal and nonnormal search strategies $\langle T_n \rangle$.

PROOF. Let H_p be the sequence of heights determined by $H_0 := 0$, $H_1 := 2^5$ and $H_p := 2^{H_{p-1}}$ for $p > 1$, so that

$$\log H_p = H_{p-1}.$$

Define

$$m(k) := \begin{cases} 2^{k-H_{p-1}}, & \text{if } H_{p-1} < k \le 3H_{p-1}, \\ \left\lfloor \left(\dfrac{H_p}{k - 3H_{p-1}} \right)^2 \right\rfloor, & \text{if } 3H_{p-1} < k \le H_p, \end{cases}$$

and $n(k) := \sum_{j=1}^{k} m(j)$.

Summing, we find that there is a constant C such that for all p and for $4H_{p-1} < k \le H_p$,

$$H_p^2/C \le n(k) \le CH_p^2 \tag{5.2}$$

and

$$H_p^2/(Ck^2) \le m(k) \le CH_p^2/k^2,$$

so that

$$1/(Ck)^2 \le m(k)/n(k) \le (C/k)^2. \tag{5.3}$$

Moreover,

$$m(k)/n(k) = \Omega(1/k^2) \tag{5.4}$$

holds for all k.

Relation (5.4) implies (1.5). For

$$h\frac{n(h) - n(\lfloor h/2 \rfloor)}{n(\lfloor h/2 \rfloor)} \ge h \sum_{h/2<k\le h} m(k)/n(k) \gg h \sum_{h/2<k\le h} 1/k^2 = \Omega(1).$$

Let C' be such that

$$h\frac{n(h) - n(\lfloor h/2 \rfloor)}{n(\lfloor h/2 \rfloor)} \ge C'.$$

Then $n(h)/n(\lfloor h/2 \rfloor) \ge 1 + C'/h$, whence $n(\lfloor h/2 \rfloor)/n(h) \le 1 - C'/(2h)$ for large h, whence

$$h\frac{n(h) - n(\lfloor h/2 \rfloor)}{n(h)} \ge \frac{C'}{2}$$

for large h, which is (1.5).

At the same time, (5.2) and (5.3) imply that (1.3) fails. For (5.3) implies $\log(n(k)/n(k-1)) \le 2(C/k)^2$ for all $4H_{p-1} < k \le H_p$, and thus

$$\sum_{k=4H_{p-1}}^{H_p} k^3 \log \frac{n(k)}{n(k-1)} \ll \sum_{k=4H_{p-1}}^{H_p} k \ll H_p^2.$$

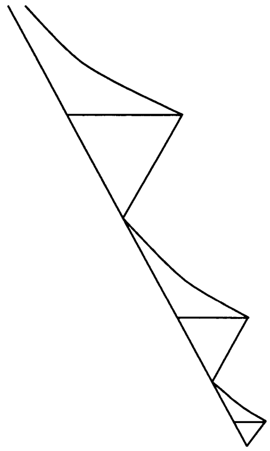

FIG. 3. *The shape with a normal strategy.*

Also,

$$\sum_{k=H_{p-1}+1}^{4H_{p-1}} k^3 \log \frac{n(k)}{n(k-1)} \le \sum_{k=H_{p-1}+1}^{4H_{p-1}} k^3 \log 2 \ll H_{p-1}^4.$$

Therefore,

$$D(H_P) \ll \log^4 n(H_P) + \sum_{p=1}^{P} \sum_{k=H_{p-1}+1}^{H_p} k^3 \log \frac{n(k)}{n(k-1)} \ll H_P^2$$

by (5.2). Thus, (1.3) fails. It is easily checked that $m(k) \le 2^k$, whence by Proposition 5.1, there is a nonnormal strategy with growth rate $\langle m(k) \rangle$, namely the depth-first strategy described in the proof of the proposition.

On the other hand, $\langle m(k) \rangle$ is also consistent with the following breadth-first strategy $\langle T_n \rangle$: for $n(H_{p-1}) < i < n(H_{p-1}) + H_p^2$, perform a breadth-first binary search as in Example 2.2, filling a complete subtree of $H_p^2 - 1$ internal nodes; this subtree starts with the root at the leftmost node of $T_{n(H_{p-1})}$ at level H_{p-1} and has its external nodes at level $3H_{p-1}$. Starting with $i = n(H_{p-1}) + H_p^2$, convert $m(3H_{p-1})$ nodes at level $3H_{p-1}$, then convert $m(3H_{p-1}+1)$ nodes at level $3H_{p-1}+1$, and so on until level H_p is reached. This can always be accomplished, since there are $2m(k-1) \ge m(k)$ external nodes available at level k. For later use, we note that in fact, except for $H_{p-1} \le k \le 3H_{p-1}$, there remain $\Omega(m(k))$ external nodes of T_i at level k for all $i \ge n(k)$.

We claim that this strategy satisfies $s_n^2/h_n^2 = \Omega(\log\log\log h_n)$, whence is normal. For any i, define $p(i)$ to be the smallest integer such that $h_i \le H_{p(i)}$.

The mean level of the external nodes of T_i is

$$\mu_i \le 3H_{p(i)-1} + \frac{\sum_{k=3H_{p(i)-1}+1}^{H_{p(i)}} \left(\frac{H_{p(i)}}{k-3H_{p(i)-1}}\right)^2 k}{\sum_{k=3H_{p(i)-1}+1}^{H_{p(i)}} \left(\frac{H_{p(i)}}{k-3H_{p(i)-1}}\right)^2}$$
$$\le 6H_{p(i)-1} + \log H_{p(i)} = 7H_{p(i)-1}.$$

Thus, the tree variance is

$$\sigma_i^2 \ge \sum_{k=9H_{p(i)-1}}^{h_i} (k-7H_{p(i)-1})^2 \frac{m(k)}{i} \ge \frac{1}{2i} \sum_{k=9H_{p(i)-1}}^{h_i} \left(\frac{k-7H_{p(i)-1}}{k-3H_{p(i)-1}}\right)^2 H_{p(i)}^2$$
$$= \Omega(h_i - 9H_{p(i)-1})$$

since $i \le n(H_{p(i)}) = O(H_{p(i)}^2)$ by (5.2).

Summing over i, we find that

$$s_n^2 \ge \sum_{\{i;\, 10H_{p(n)-1} \le h_i \le h_n\}} \sigma_i^2 \gg \sum_{k=10H_{p(n)-1}}^{h_n} km(k)$$
$$\asymp \sum_{k=10H_{p(n)-1}}^{h_n} k(H_{p(n)}/k)^2 \asymp H_{p(n)}^2 \log \frac{h_n}{10H_{p(n)-1}}$$
$$\ge h_n^2 \log \frac{h_n}{10H_{p(n)-1}}.$$

This gives $s_{n(H_{p-1})}^2 / H_{p-1}^2 \gg \log(H_{p-1}/10H_{p-2}) \gg \log H_{p-1}$, whence for $n(H_{p-1}) \le n \le n(H_{p-1}(\log H_{p-1})^{1/4})$,

$$s_n^2/h_n^2 \ge s_{n(H_{p-1})}^2/h_n^2 \gg (\log H_{p-1})^{1/2} \gg \log\log\log h_n.$$

In addition, for $n(H_{p-1}(\log H_{p-1})^{1/4}) \le n \le n(H_p)$, we have

$$s_n^2/h_n^2 \gg \log \frac{h_n}{10H_{p-1}} \gg \log\log H_{p-1} \gg \log\log\log h_n.$$

Putting these together, we get $s_n^2/h_n^2 = \Omega(\log\log\log h_n)$, as claimed. □

6. A small-variance normal search. We showed in Proposition 1.7 that, for searches satisfying the regularity condition (1.5), normality is equivalent to the condition (1.1) that variance grow faster than the square of the height. We conclude now by giving an interesting example that shows that without this regularity condition, normal searches may exhibit extremely small growth of variance.

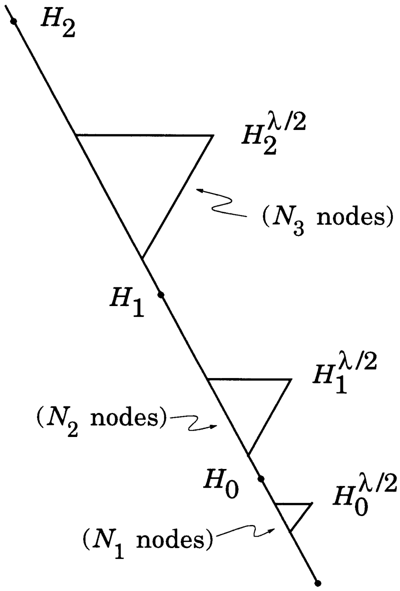

FIG. 4. *The shape for a small variance, with heights indicated.*

EXAMPLE 6.1. Given any $\lambda \in (0, 1]$, there is a normal strategy with

$$\liminf_{n\to\infty} s_n/n^{\lambda/6} = \liminf_{n\to\infty} s_n/h_n^{\lambda} = 0. \tag{6.1}$$

PROOF. Choose an integer $r \in (3/\lambda + 1/2, 6/\lambda + 1)$. Choose H_0 to be a power of 2 so large that $H_0^{\lambda/2} \geq \log H_0^{6r}$. Define inductively $H_{j+1} := H_j^r$ and $N_j := H_j^6$. The strategy is the following hybrid of linear search and depth-first binary search, that is, of Examples 2.1 and 2.3.

For the first H_0 steps, do a linear search: form T_i from T_{i-1} by converting the leftmost external node to an internal node. Thus, the height of T_i is $i-1$ in this range. For the next $N_1 - \log N_1 - 1$ steps, do a depth-first binary search: convert the leftmost external node with level lying in the range

$$\big(\lfloor H_0^{\lambda/2} \rfloor - \log N_1, \lfloor H_0^{\lambda/2} \rfloor\big)$$

to an internal node. Note that the heights of these trees are all H_0. Now continue with a linear search up to height H_1 by always converting the leftmost external node to an internal node. Then do a depth-first binary search that fills in the levels

$$\big(\lfloor H_1^{\lambda/2} \rfloor - \log N_2, \lfloor H_1^{\lambda/2} \rfloor\big).$$

Continue in this fashion forever, alternating linear search up to height H_j with binary search of N_{j+1} nodes inserted at height approximately $(H_j)^{\lambda/2}$.

Recalling that $n(h) := \max\{n;\ h_n = h\}$, define $n'(h) := \min\{n;\ h_n = h\}$. We shall first establish that

$$s^2_{n'(H_j)} \asymp H_j^{(6+\lambda)/r} \log H_j, \tag{6.2}$$

whence the equations

$$H_j^\lambda = o(s^2_{n'(H_j)}) \tag{6.3}$$

$$s^2_{n'(H_j)} = o(H_j^{2\lambda}) \tag{6.4}$$

follow by our choice of r. Using (6.3), we shall then establish that

$$\tilde{s}_{n,\varepsilon} = o(1). \tag{6.5}$$

Equation (6.4) implies (6.1), while (6.5) implies asymptotic normality by Proposition 1.1.

To prove the first claim, (6.2), note that during the steps of the $(j+1)$st depth-first binary search, the set of external nodes is approximately the union of H_j nodes, one at each level; N_j nodes at level $\lfloor H_{j-1}^{\lambda/2} \rfloor$; and some nodes at level $\lfloor H_j^{\lambda/2} \rfloor$. (Since $\log N_l$ increases geometrically, we may neglect the nodes from the previous binary searches for the purpose of estimating the variance.) As in Example 2.3, an upper bound for the tree variance σ_i^2 is the mean squared distance to level $\lfloor H_j^{\lambda/2} \rfloor$, which is

$$\ll (H_j^3 + \lfloor H_j^{\lambda/2} \rfloor^2 N_j)/i \asymp H_j^{6+\lambda}/i. \tag{6.6}$$

After adding the first N_j nodes of the $(j+1)$st depth-first binary search, this is also an approximate lower bound, since the mean level is then at least $H_j^{\lambda/2}/2$. Therefore, the total variance added during these steps is

$$\asymp H_j^{6+\lambda} \sum_{i=N_j}^{N_{j+1}} 1/i \asymp H_j^{6+\lambda} \log H_j.$$

Similarly, the variance added during the steps of the linear search from height $H_j + 1$ to height H_{j+1} is

$$\asymp (H_{j+1}^3 + H_j^\lambda N_j) \sum_{i=N_{j+1}}^{N_{j+1}+H_{j+1}} 1/i \quad \asymp \quad H_j^{3r-5r} = o(1). \tag{6.7}$$

Hence,

$$s^2_{n'(H_{j+1})} \asymp s^2_{n(H_j)} \asymp H_j^{6+\lambda} \log H_j \quad \asymp \quad H_{j+1}^{(6+\lambda)/r} \log H_{j+1},$$

which establishes the first claim.

It remains to establish the second claim, (6.5). Fix $\varepsilon > 0$. Given n, there is a unique j such that $n'(H_j) \le n < n'(H_{j+1})$. By virtue of (6.3), for large enough n, we have

$$\varepsilon s_n \ge \varepsilon s_{n'(H_j)} > \lfloor H_j^{\lambda/2} \rfloor.$$

This means that the only nodes that contribute to $\tilde{s}_{n,\varepsilon}$ are those at levels greater than $\lfloor H_j^{\lambda/2} \rfloor$. That is, *only nodes arising from the last one or two linear searches contribute to* $\tilde{s}_{n,\varepsilon}$.

From (6.7), we see that variance added to $\tilde{s}_{n,\varepsilon}$ during linear steps is $o(s_n^2)$. We must show that variance added during binary steps is also $o(s_n^2)$. We need only consider the last binary search, at level $H_j^{\lambda/2}$. The variance arising from early binary searches is zero, since at this stage there are no nodes at level greater than $H_j^{\lambda/2}$.

Consider this final binary search. Fixing n, define

$$\tilde{\sigma}_{i,\varepsilon}^2 := \mathbf{E}\big[(X_i - \mathbf{E}[X_i])^2;\ |X_i - \mathbf{E}[X_i]| > \varepsilon s_n\big],$$

so that $s_{n,\varepsilon}^2 = \sum_{i=1}^n \tilde{\sigma}_{i,\varepsilon}^2$. Because only linear nodes contribute, the variance $\tilde{\sigma}_{i,\varepsilon}^2$ for step i is bounded, not by (6.6), but by H_j^3/i. Since $i \geq N_j$, the total contribution from the first H_j^3 steps is thus $\asymp H_j^6/N_j = 1 = o(s_n^2)$, as desired. On subsequent binary steps, on the other hand, an asymptotic lower bound for σ_i^2 is $\lfloor H_j^{\lambda/2} \rfloor^2 H_j^3/i$. More generally, an asymptotic lower bound for σ_i^2 after L steps of this binary search is $H_j^\lambda \min\{L, N_j\}/i$. For $L < N_j$, the mean is at most $\asymp H_j^{\lambda/2}/2$, so we get variance of $(H_j^{\lambda/2}/2)^2$ from each of L nodes. For $L > N_j$, the mean is greater than or equal to $\asymp H_j^{\lambda/2}/2$, so we get variance of $(H_j^{\lambda/2}/2)^2$ from each of the N_j nodes from the just previous binary search. Since $\tilde{\sigma}_{i,\varepsilon}^2$ has asymptotic upper bound of H_j^3/i, while σ_i^2 has asyptotic lower bound of $H_j^3 H_j^\lambda/i$, we have $\tilde{\sigma}_{i,\varepsilon}^2 = o(\sigma_i^2)$ for these steps. Summing, we find that their total contribution is also $o(s_n^2)$, verifying (6.5) and completing the proof. □

Acknowledgments. We are grateful to Janice Lent and Hosam Mahmoud for clarifying the role of these models in practice. We thank one of the referees for finding some minor errors in a previous version of the manuscript.

REFERENCES

[1] FELLER, W. (1971). *An Introduction to Probability Theory and Its Applications* **2**, 2nd ed. Wiley, New York.

[2] LENT, J. H. and MAHMOUD, H. M. (1996). On tree-growing search strategies. *Ann. Appl. Probab.* **6** 1284–1302.

[3] MAHMOUD, H. M. (1992). *Evolution of Random Search Trees.* Wiley, New York.

DEPARTMENT OF MATHEMATICS
INDIANA UNIVERSITY
BLOOMINGTON, INDIANA 47405-5701
E-MAIL: rdlyons@indiana.edu
kzumbrun@indiana.edu