On Random Cartesian Trees

Luc Devroye*
School of Computer Science, McGill University, 3480 University Street, H3A 2A7, Montreal, Quebec, Canada

ABSTRACT

Cartesian trees are binary search trees in which the nodes exhibit the heap property according to a second (priority) key. If the search key and the priority key are independent, and the tree is built based on n independent copies, Cartesian trees basically behave like ordinary random binary search trees. In this article, we analyze the expected behavior when the keys are dependent: in most cases, the expected search, insertion, and deletion times are $\Theta(\sqrt{n})$. We indicate how these results can be used in the analysis of divide-and-conquer algorithms for maximal vectors and convex hulls. Finally, we look at distributions for which the expected time per operation grows like n^a for $a \in [1/2, 1)$.

1. INTRODUCTION

Cartesian trees were introduced by Vuillemin [40, 41] as a data structure for storing data according to two keys: they are binary search trees with respect to the first key, and the nodes have the heap property with respect to the second key. The minimal second key is found at the root. Cartesian trees can thus also be used as priority queues. Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be the values of these pairs of keys. It is clear that this sequence uniquely determines the structure of the binary tree. Permuting the pairs in the sequence does not alter the Cartesian tree. To visualize things, it is helpful to put the (X_i, Y_i) pairs in their exact position in $\mathbb{R}^2$ and draw the tree, as in Figure 1.

* Research was sponsored by NSERC Grant A3456 and by FCAR Grant 90-ER-0291.

Random Structures and Algorithms, Vol. 5, No. 2 (1994)
 CCC 1042–9832/94/**020305–23**

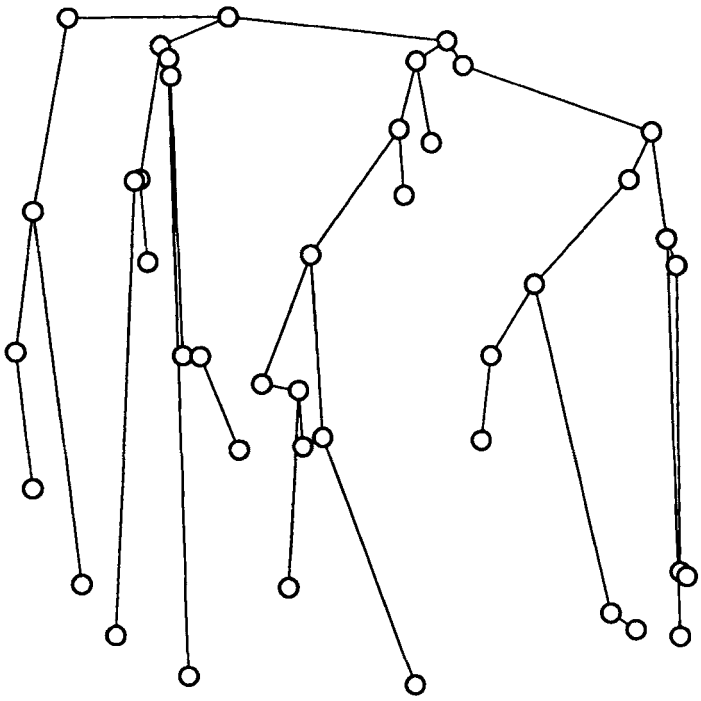

Fig. 1. Random Cartesian tree, in which points are placed at their true x and y coordinates. The y-axis points down.

We analyze the properties of the Cartesian tree when the pairs of keys are independent and identically distributed (i.i.d.). The distribution of (X, Y) influences the shape of the tree as well as the complexity of the operations. In fact, we will show that for most distributions of (X, Y), theexpected time complexity of ordinary dictionary or priority queue operations becomes an unacceptable $\Theta(\sqrt{n})$. It makes sense to assume that (X, Y) has a joint density. Consider, for example, a distribution in which all points fall on the perimeter of a circle or polygon with at least five vertices. It is easy to see that the Cartesian tree has expected height $\Omega(n)$. Thus, throughout the article, we assume that the pair (X, Y) has an absolutely continuous distribution, so that events like $X_i = X_j$ or $Y_k = Y_l$ occur with zero probability. A few special data structures are worth mentioning:

A. Ordinary Binary search Trees under the operations INSERT and SEARCH [10] can be considered as Cartesian trees in which the second key is the time of insertion: elements down any path in the tree have increasing time stamps. Of course, the second key is not usually stored, thus causing the analogy with Cartesian trees to break down as soon as deletions, splay operations, rotations, or other balancing operations are performed.

B. Treaps ([41]; see also [2]) are Cartesian trees used for dictionary operations only, in which a second key is generated at random and independently of the first key. By explicitly storing the second key, deletions transform Cartesian trees into Cartesian trees. Treaps are distributed and shaped like random binary search trees under the equiprobable random

permutation model. While deletions and insertions take $O(\log n)$ time on the average, their implementations are of course different from those in binary search trees.

C. Pagodas were first introduced by Françon, Viennot; and Vuillemin [20] as an alternative for a priority queue. Barring certain technical modifications, the pagodas can be thought of as Cartesian trees in which the first key is a time stamp, i.e., the time of insertion of an element. The properties are good on the average if elements are inserted in random order. In fact, if the second keys form an i.i.d. sequence, pagodas are distributed as random binary search trees under the equiprobable random permutation model.

D. Randomized Pagodas are pagodas in which the first key is drawn independently from a fixed distribution. This insures the random binary search tree distribution regardless of how the second keys are picked.

E. Priority search Trees [34] are not Cartesian trees although they too are designed to store information for double use as a dictionary and a priority queue. This is achieved by creating a binary search tree with respect to the first key and by adding pointers to the nodes with the largest second keys in the subtrees of all nodes.

2. OPERATIONS ON CARTESIAN TREES

The quantities that interest us are those that describe the complexity of the dictionary and priority queue operations. A quick revision of these operations is therefore in order.

SEARCH

Searching for an element with first coordinate X_i takes time proportional to the path distance between the root and the element. For element (X_i, Y_i) in a tree of size n, this distance is called the depth $D_{n,i}$. The maximal such distance is called the height:

$$H_n = \max_{1 \le i \le n} D_{n,i} .$$

DELETE THE ROOT

Deleting the node with the smallest Y-value is like deleting the root. Look at the tree as in Figure 2, where we call the left spine the chain of nodes starting with the leftchild of the root and taking all the right children down the tree. The right spine is defined symmetrically. To delete the root, we only have to reconnect the left and right spines according to increasing Y_i values, very much as in a merge of two sorted lists. The time taken by this is proportional to the sum of the lengths of the left and right spines. Let I be the index of the root (X_I, Y_I), and define for general i the quantity $D^*_{n,i}$, which is the depth of (X_i, ∞) in the Cartesian tree holding (X_j, Y_j), $1 \le j \le n$, in which Y_i is replaced by ∞. It is easy to see that $D^*_{n,I}$ is equal to the sum of the lengths of the two spines, plus one, as (X_I, ∞) would be a leaf dangling at the bottom of the merged spines. Thus, the DELETE operation requires time proportional to $D^*_{n,I}$. (See Fig. 3.)

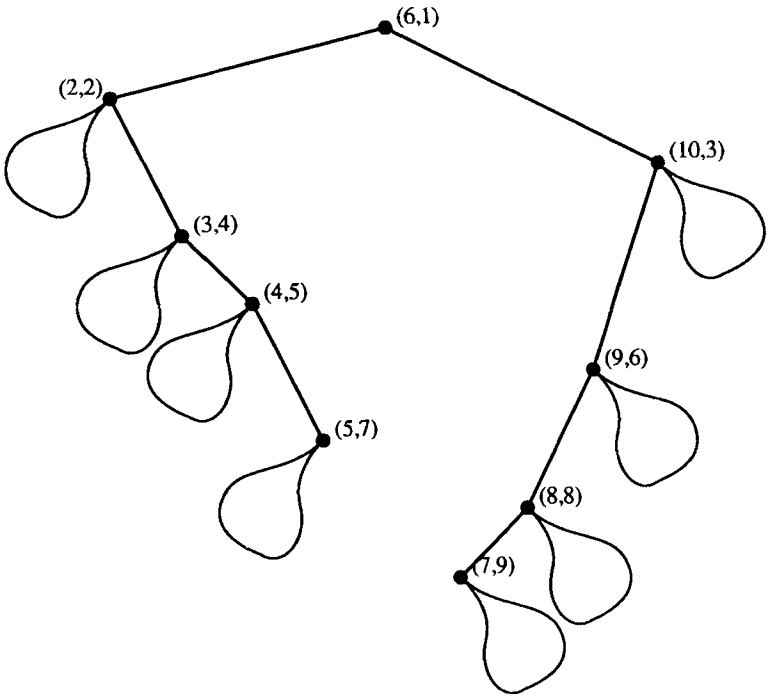

Fig. 2. Cartesian tree with root, left spine, and right spine. Subtrees are shown as amorphous blobs; y-axis points down.

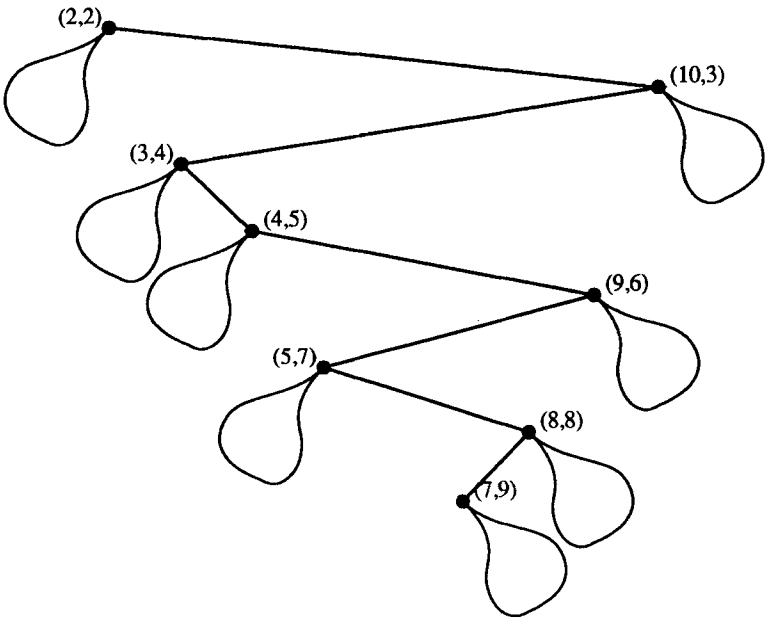

Fig. 3. Cartesian tree after deletion of root: left and right spines are merged. The y-axis points down.

DELETE

The deletion of (X_i, Y_i) is done in two stages. The first stage reduces to a search for (X_i, Y_i). Then, if T is the subtree rooted at (X_i, Y_i), we proceed with a delete-root operation as explained above, as only T is affected by the deletion. Recalling a crucial property from the delete-root operation, we see that deleting the i-th pair takes time proportional to $D^*_{n,i}$. (See Fig. 4.)

INSERT

The INSERT operation is the reverse of the delete operation. Again, we proceed in two stages. Let T be the original n-node Cartesian tree, and let T' be a copy of T in which we keep only those nodes with Y-value less than Y_{n+1} (so T' is the top part of T). In a first stage, we insert the new element (X_{n+1}, Y_{n+1}) in T'. Then we overlay T and T'. A collision occurs if we can't do this without creating a node (the parent of (X_{n+1}, Y_{n+1})) with two left children or two right children, one of them of course being (X_{n+1}, Y_{n+1}). Let T'' be the subtree rooted at that other child. In the second stage, we merge the new element and T'' by making it the root of the subtree (since it has the smallest Y-value). T'' is split into T_l and T_r, where T_l contains all elements with X-value less than X_{n+1}, and T_r contains the other elements of T. This split creates the spines of Figure 2 from the tree of Figure 3. The new element is made the root by connecting the spines to it as in Figure 2. It is noteworthy that the time taken by the entire INSERT operation is proportional to the distance between the root of T and the position of (X_{n+1}, ∞), which is $D^*_{n+1,n+1}$.

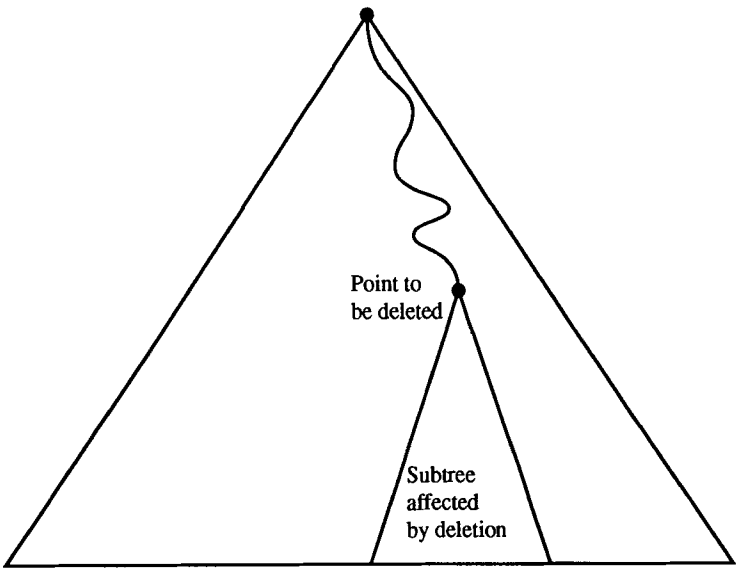

Fig. 4. To delete the marked point from a Cartesian tree, apply a delete-root operation to its tree; y-axis points down.

OTHER OPERATIONS

In pagodas, all INSERT operations are done for elements having the largest X-value thus far. In ordinary binary search trees, all inserts are for elements possessing the largest Y-value seen thus far. The time taken by these operations is reduced in a simple manner to the quantities introduced thus far. Without enlarging our model it is not possible to meaningfully discuss the operation DECREASEKEY [38].

We retain from this brief introduction that the quantities of interest to us are $D_{n,i}$, $D^*_{n,I}$, $D^*_{n,i}$ and H_n. We also introduce the worst-case insertion or deletion time

$$H^*_n = \max_{1 \le i \le n} D^*_{n,i} .$$

Observe that $D_{n,i} \le D^*_{n,i}$ and $H_n \le H^*_n$.

3. THE INDEPENDENT MODEL

In this section, we briefly review what is known for the independent model, i.e., the model in which Y is independent of X. Under this model, the random Cartesian tree is distributed as a random binary search tree under the equiprobable random permutation model, i.e., a binary search tree on n nodes obtained by inserting, in the standard manner, the values $\sigma_1, \ldots, \sigma_n$ of a random permutation of $\{1, \ldots, n\}$ into an initially empty tree. Equivalently, the search tree is obtained by inserting n i.i.d. uniform $[0, 1]$ random variables $X_1, \ldots, X_n$. We refer to such a tree simply as a RANDOM BINARY SEARCH TREE.

Thus, $D^*_{n,I}$ and the $D^*_{n,i}$'s are distributed as the depth of the last node inserted in a random binary search tree (for $D^*_{n,I}$, this is a corollary of the independence of the X and Y coordinates). It is known that the expected depth of the nth node in a random binary search tree on n nodes is asymptotic to $2 \log n$ in many senses [1, 26]. The limit law of the depth of this node, and various other properties were obtained in [25, 26, 37, 35, 33, 24; 28, 17]. Various connections with the theory of random permutations [37] and the theory of records [17] were pointed out over the years. Thus, from [17]:

Lemma 1. *Every $D^*_{n,i}$ in a random Cartesian tree under the independent model is distributed as L_n, the depth of the last node inserted in a random binary search tree. Also,*

$$\mathbf{E}\{L_n\} = 2 \sum_{i=1}^{n} \frac{1}{i} - 2 ,$$

$$\mathrm{Var}(L_n) = 2 \sum_{i=1}^{n} \frac{1}{i} - 4 \sum_{i=1}^{n} \frac{1}{i^2} + 2 ,$$

$$\frac{L_n}{\mathbf{E}\{L_n\}} \to 1 \quad \text{in probability as} \quad n \to \infty ,$$

and

$$\frac{L_n - \mathbf{E}\{L_n\}}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{\mathcal{L}} N ,$$

where $\xrightarrow{\mathcal{L}}$ *denotes convergence in distribution, and N is a standard normal random variable.*

The exact distribution of L_n was derived by Lynch [29] and Knuth [26] (see also [37, p. 144]). We next turn to $D_{n,i}$. Clearly, $D_{n,i} \leq D^*_{n,i}$, so that Lemma 1 describes already part of the story. In fact, $D_{n,i}$ is very close to $D^*_{n,i}$. We have the following distributional property: $D_{n,i}$ is distributed as L_N, where N is uniformly distributed on $\{1, \ldots, n\}$ and is independent of the tree, and L_k is the depth of the last node in a random binary search tree. As already noted by Françon, Viennot, and Vuillemin [20]; this immediately yields the following:

$$\begin{aligned}\mathbf{E}L_N &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}L_i = \frac{1}{n}\sum_{i=2}^{n}\sum_{j=2}^{i}\frac{2}{j} \text{ (Lemma 1)}\\ &= \frac{n+1}{n}\sum_{i=2}^{n}\frac{2}{i} - \frac{2(n-1)}{n} \sim 2\log n\,.\end{aligned}$$

Furthermore,

$$\begin{aligned}\mathrm{Var}\{L_N\} &= \mathrm{Var}\{\mathbf{E}\{L_N \mid N\}\} + \mathbf{E}\{\mathrm{Var}\{L_N \mid N\}\}\\ &= O(1) + \mathbf{E}\left(I_{N>1}\sum_{i=2}^{N}\frac{2}{i}\left(1-\frac{2}{i}\right)\right)\\ &\sim 2\log n\,.\end{aligned}$$

Thus, $L_N/\mathbf{E}L_N \to 1$ in probability, by Chebyshev's inequality. With a little extra work, one can also show that

$$\frac{L_N - \mathbf{E}L_N}{\sqrt{\mathrm{Var}\{L_N\}}} \xrightarrow{\mathcal{L}} \text{ the normal distribution}\,.$$

Next, observe that H_n is distributed as the height of a random binary search tree. Thus, from Devroye [14, 16]:

Lemma 2. *Let H_n be the height of a random Cartesian tree under the independent model. Then*

$$\frac{H_n}{c\log(n)} \to 1 \quad \text{in probability}$$

and

$$\mathbf{E}\{H_n\} \sim c\log(n)$$

where $c = 4.31107\ldots$ *is the solution of*

$$c\log\left(\frac{2e}{c}\right) = 1\,; \quad c \geq 2\,.$$

Remark 1. For the random binary search tree, Robson [36] and Pittel [35] provided the first analyses of H_n. Surveys of known results can be found in [39, 22, and 31]. In [14, 16], the theory of branching processes is used in the analysis of H_n. Flajolet and Odlyzko [19] studied H_n under other models of randomization. ∎

This brings us finally to H_n^*. Using Lemma 2 and some large deviation inequalities, we have:

Lemma 3. *Let H_n^* be the worst-case insertion time of a random Cartesian tree under the independent model. Then*

$$\frac{H_n^*}{c\log(n)} \to 1 \quad \text{in probability}$$

and $\mathbf{E}\{H_n^*\} \sim c\log(n)$, *where* $c = 4.31107\ldots$ *is as in Lemma* 2.

Proof. We have $H_n^* \geq H_n$, so that lower bounds for H_n^* can be derived from Lemma 2. It suffices, therefore, to look at upper bounds only. Let $t > 0$ be arbitrary. Bonferroni's inequality implies the following:

$$\mathbf{P}\{H_n^* > t\} \leq \sum_{i=1}^{n} \mathbf{P}\{D_{n,i}^* > t\} = n\mathbf{P}\{L_n > t\} = n\mathbf{P}\left\{\sum_{j=2}^{n} Z_j > t\right\}$$

where $Z_2, \ldots, Z_n$ are independent Bernoulli random variables taking the value one with probability $2/2, \ldots, 2/n$, respectively [17]. We use Chernoff's exponential bounding technique [7]: let $S = \sum_{i=2}^{n} (1/j)$ and let $\lambda > 0$ be a constant to be picked further on. Then

$$\begin{aligned}
\mathbf{P}\{H_n^* > t\} &\leq n\mathbf{E}\exp\left(-\lambda t + \sum_{i=2}^{n} \lambda Z_i\right) \\
&= ne^{-\lambda t}\prod_{i=2}^{n}\left(1 - \frac{2}{i} + \frac{2e^{\lambda}}{i}\right) \\
&\leq n\exp(-\lambda t + 2S(e^{\lambda} - 1))\,.
\end{aligned}$$

Take $\lambda = \log(t/(2S))$ and note that the upper bound becomes, with $t = uS$ for constant $u > 2$:

$$\begin{aligned}
\mathbf{P}\{H_n^* > t\} &\leq n\exp(t - 2S - t\log(t/(2S))) \\
&= \exp((1 + o(1))\log n \times (u - 1 - u\log(u/2))) \\
&\to 0
\end{aligned}$$

as $n \to \infty$ when $u > c$. The claim about $\mathbf{E}H_n^*$ follows easily from the bound given above and the fact that $\mathbf{E}H_n^* = \int_0^{\infty} \mathbf{P}\{H_n^* > t\}\,dt$. ∎

Remark 2. If (X, Y) has a density f, then under some circumstances, we still retain the $\log n$ expected behavior for the quantities studied in Lemmas 1 through 3. A sufficient condition is that f is supported on $[0, 1]^2$ and that there exist positive constants a, b such that $b \geq f \geq a > 0$ on $[0, 1]^2$. This condition is rather restrictive, and we won't pursue this angle here.

4. THE GENERAL MODEL

Assume that the prototype pair (X, Y) has a density f on $\mathbb{R}^2$. The properties of the random Cartesian tree deteriorate quickly with the amount of dependence between X and Y. This leads us first of all to some measure of the dependence between two random variables X and Y. Many such measures have been proposed in the statistical literature, but we can't afford to choose: we have to use the measure that imposes itself in a natural fashion. The following quantity, which we dub the DOMINATION FACTOR, seems to capture what we want:

$$d \stackrel{\text{def}}{=} \inf_{Z,W} \sup_{A,B} \frac{\mathbf{P}\{X \in A, Y \in B\}}{\mathbf{P}\{Z \in A\}\mathbf{P}\{W \in B\}} .$$

Here the infimum is over all random variables Z and W and the supremum is over all Borel sets A and B. Because X and Y are possibly dependent, one may ask what the "closest" independent pair (Z, W) looks like, where closest is defined in terms of the minimization of d in the inequality

$$\mathbf{P}\{X \in A, Y \in B\} \leq d\mathbf{P}\{Z \in A\}\mathbf{P}\{W \in B\} .$$

Remark 3. Since (X, Y) has a density f, it is sufficient to consider only Z and W with densities g and h (say). Then we have

$$d \stackrel{\text{def}}{=} \inf_{g,h} \operatorname*{ess\,sup}_{(x,y)} \frac{f(x, y)}{g(x)h(y)} ,$$

where ess sup denotes the essential supremum with respect to f. To see why this is true, we provide a brief explanation. Let μ and $\nu = \nu_1 \times \nu_2$ denote the probability measures for (X, Y) and (Z, W), respectively. If μ is not absolutely continuous with respect to ν, then there exists a set $C_1 \times C_2$ such that $\nu_1(C_1)\nu_2(C_2) = 0$, yet $\mu(C_1 \times C_2) > 0$. For this choice of ν, we note that the supremum in the definition of d is infinite. Therefore, the infimum over ν is certainly reached with respect to product measures ν such that μ is absolutely continuous with respect to ν. So assume that $\nu = \nu_{ac} + \nu_s$, its decomposition into a part that is absolutely continuous with respect to μ, and its singular part. Let the probability weights of these parts be p and $1 - p$, respectively. The supremum in the definition of d remains the same even if we restrict ourselves to sets A, B for which $\nu_s(A \times B) = 0$. But in that case, the measure ν_{ac}/p makes the infimum in the definition of d smaller. So we may assume without loss of generality that ν is also absolutely continuous with respect to μ, and therefore that ν is induced by some product density $g \times h$. But

$$\int_{A\times B} f = \int_{A\times B} \frac{f}{gh}\, gh \leq \operatorname*{ess\,sup}_{x,y} \frac{f(x, y)}{g(x)h(y)} \times \int_A g \int_B h ,$$

thus showing that for every choice of (g, h),

$$\sup_{A,B} \frac{\int_{A\times B} f}{\int_A g \int_B h} \leq \operatorname*{ess\,sup}_{x,y} \frac{f(x, y)}{g(x)h(y)}$$

Let $S_{x,u}$ denote the closed sphere of radius u centered at x. Then also

$$\sup_{A,B} \frac{\int_{A\times B} f}{\int_A g \int_B h} \geq \operatorname*{ess\,sup}_{x,y} \liminf_{u\to 0} \frac{\int_{S_{x,u}\times S_{y,u}} f}{\int_{S_{x,u}} g \int_{S_{y,u}} h} = \operatorname*{ess\,sup}_{x,y} \frac{f(x, y)}{g(x)h(y)},$$

where we used the Lebesgue density theorem [42, p. 100]). This concludes Remark 3.

It is clear that $d \geq 1$ (otherwise, derive a contradiction by integrating f over $\mathbb{R}^2$). In fact, $d = 1$ if and only if f can be decomposed as $f(x, y) = g(x)h(y)$, in which case we see that X and Y are independent. It that case, we can take $Z = X$ and $W = Y$.

Example 1. There is no reason why the choice $Z = X$ and $W = Y$ would lead to the infimum in the definition of d. In fact, often it does not. Consider f as the uniform density on the unit circle of $\mathbb{R}^2$. Then $d = 4/\pi$: indeed, for Z and W both uniform on $[-1, 1]$, we see that f/gh takes the value $4/\pi$ on the unit circle, and the value zero elsewhere. Thus, $d \leq 4/\pi$. Also, for any densities g and h on $[-1, 1]$, we have

$$\sup_{x,y} \frac{f(x, y)}{g(x)h(y)} \geq \frac{1}{\pi \inf_{x\in[-1,1]} g(x) \inf_{y\in[-1,1]} h(y)} \geq \frac{4}{\pi}.$$

We conclude that $d = 4/\pi$. The optimizing random variables Z and W are thus not distributed like the marginal random variables X and Y.

Example 2. If f is a bounded density on any compact set $C \subseteq [a, b] \times [a', b']$, then d is finite, and, in fact, $d \leq \|f\|_\infty (b - a)(b' - a')$. If f is the uniform density on the same compact set, and $C \subseteq C_1 \times C_2$, where C_1 and C_2 are the smallest closed sets with the properties that $\mathbf{P}\{X \in C_1\} = 1$, and $\mathbf{P}\{Y \in C_2\} = 1$, then $d = \lambda(C_1 \times C_2)/\lambda(C)$, where $\lambda(\cdot)$ denotes Lebesgue measure on $\mathbb{R}^2$.

Example 3. Another important family of distributions occurs when f is radially symmetric, so that

$$f(x, y) = F(\sqrt{x^2 + y^2})$$

for some function F. Assume that F is nonincreasing. Then,

$$F(\sqrt{x^2 + y^2}) \leq \sqrt{F(|x|)}\sqrt{F(|y|)}$$

for all x and y. If we take $g(x) = h(x) = C\sqrt{F(|x|)}$, where C is a normalization constant, we obtain that

$$d \le \left(\int \sqrt{F(|x|)}\, dx \right)^2 .$$

Thus, for most radially symmetric densities, d is finite. Note that F can be unbounded at the origin.

Example 4. The value of d is invariant under strictly monotone transformations of the axes. That is, the value for (X, Y) is the same as for $(G(X), H(Y))$, where G and H are strictly monotone transformations. Thus, we can always assume, if it is convenient to do so, that X and Y have support on $[0, 1]$. In this respect, d measures a deeply rooted type of dependence.

Example 5. Consider a bivariate normal distribution with correlation coefficient $\rho \in [-1, 1]$. By the previous example, we can assume that the variances of X and Y are one—otherwise, just rescale everything, leaving d unaltered. By the same token, assume that the means are zero. If g and h are standard normal densities, then

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} (x^2 + y^2 - 2\rho xy) \right) \le \frac{1}{\sqrt{1-\rho^2}}\, g(x)h(y) .$$

This inequality implies that

$$d \le \frac{1}{\sqrt{1-\rho^2}} .$$

Not unexpectedly, there is a one-to-one relationship between $|\rho|$ and the upper bound for d, with the bound on d varying from 1 at $\rho = 0$ to ∞ as $|\rho| \to 1$.

Example 6. It helps to construct an example in which $d = \infty$. Consider X uniform on $[0, 1]$ and $Y = X + U_1 U_2$ where U_1, U_2 are independent and uniform on $[0, 1]$ and independent of X. The joint density looks like a rollerskate slope with an infinite crest at $y = x$. [Note in particular that $U_1 U_2$ has density $-\log u$ on $(0, 1)$.] Assume $d < \infty$. Fix g and h and $x \in (0, 1)$ and let $y \uparrow x$. Then $f(x, y) \uparrow \infty$. This implies that at almost all x, $\max(g(x), h(x)) = \infty$, which leads to a contradiction. In general, examples can be constructed with $d = \infty$ based upon densities that have infinite-valued crests that are not aligned with one of the two axes.

Example 7. Bounded densities as well can have infinite values of d. Just take $f = 1$ on an unbounded set C of Lebesgue measure one, and $f = 0$ elsewhere. Assume that for every x, the Lebesgue measure of the section of C at x is positive, and similarly for every y. Then, extending Example 2, we see that $d = \infty$.

Our first main result, Theorem 1, states that both H_n and H_n^* for a random Cartesian tree are $O(\sqrt{n})$ in probability when the domination factor d is finite. In fact, the main asymptotic term in our bounds is proportional to $\sqrt{dn}$. This bound describes more the norm than the exception: in an example following Theorem 1, we establish that for the uniform distribution in the unit square, the expected

depth $\mathbf{E}D_{n1}$ is $\Omega(\sqrt{n})$ for all rotations of the square except when the axes are aligned with the coordinate axes, in which case we know that $\mathbf{E}H_n = O(\log n)$. In other words, there is a sudden jump in the performance. What matters is the collection of points near the border of the square. As they are about evenly distributed, most of these "border" points along two sides are either pure left descendants or pure right descendants from the root, and there are about $\sqrt{n}$ of these when the square is not aligned with the coordinate axes. However, when the square is perfectly aligned, very few of these extreme points are pure left or pure right descendants of the root: only those near the top two vertices play a role, and there are about $\log n$ such points.

Theorem 1. *Consider a random Cartesian tree on n nodes under the general model. Assume that the domination factor d is finite. Then, for every* $\epsilon > 0$,

$$\lim_{n\to\infty} \mathbf{P}\{H_n^* > (C+\epsilon)\sqrt{dn}\} = 0\,,$$

where $C = 4e\sqrt{\log 4}$. *Also,*

$$\limsup_{n\to\infty} \frac{\mathbf{E}\{H_n^*\}}{C\sqrt{dn}} \le 1\,.$$

The same inequalities remain valid for H_n.

Proof. The proof of Theorem 1 is combined with that of Theorem 2 below. ∎

5. LOWER BOUNDS

The bound of Theorem 1 cannot be improved for many simple distributions. To clarify this, just take the uniform distribution on the trapezoid T formed by intersecting $[0,1]^2$ with $\{(x, y)\colon x < y < x + c\}$, with $0 < c \le 1$. The area of the trapezoid is $(1/2)(1-(1-c)^2) = c - c^2/2$. Hence,

$$f(x, y) = \frac{1}{c - c^2/2}\, I_T(x, y)\,.$$

Example 2 shows that $d = 1/(c - c^2/2)$. Let L_n denote the length of the path in the tree that leads to the node with the maximal x-value. Clearly, $H_n \ge L_n$, so that lower bounds on $\mathbf{E}L_n$ provide us with lower bounds on $\mathbf{E}H_n$. We will also see further on that modulo a constant, this gives us a lower bound for every $\mathbf{E}D_{n,i}$.

We take an integer m large enough such that $1/m < c$. Then partition the unit square into a rectangular grid of m by m with sides equal to $1/m$. Mark the m grid cells that straddle the diagonal of the square. Let $E_1, \ldots, E_m$ be the indicators of the events that the marked grid cells contain at least one data point, with E_1 referring to the cell with the smallest y-values, and so on up. A simple geometric argument shows that

$$L_n \geq \sum_{i=1}^{m} E_i .$$

Hence, if the marked grid cells intersected with our trapezoid T yield the triangles $S_1, \ldots, S_m$,

$$\begin{aligned}
\mathbf{E}L_n &\geq m\mathbf{E}E_1 \\
&= m(1-(1-d\,\text{area}(S_1))^n) \\
&= m(1-(1-d/(2m^2))^n) \\
&\geq m(1-\exp(-dn/(2m^2))) \\
&\geq m/2 \\
&\geq \sqrt{dn/4\log 4}-1
\end{aligned}$$

if we choose $m = \lfloor \sqrt{dn/\log 4} \rfloor$. Recall that n has to be so large that $m = m(n) > 1/c$. Thus, we have

$$\mathbf{E}H_n \geq \sqrt{dn/4\log 4}-1 .$$

The upper bound in Theorem 1 cannot be improved upon in terms of d and n.

To make matters worse, the $\Omega(\sqrt{n})$ behavior is also inherited by every $D_{n,i}$. In the same example, consider all nodes whose x-coordinate is between 1/2 and 1. For those nodes, the path to the root necessarily visits at least one node in every triangle S_i with $i \leq m/2$ that is occupied. Thus,

$$D_{n,i} \geq I_{[X_i \geq 1/2]} \sum_{j=1}^{\lfloor m/2 \rfloor} E_j ,$$

Given that $X_i \geq 1/2$, the number of points in E_j is binomial $(n-1, d/(2m^2))$ $(j \leq m/2)$. Thus, since for $c \leq 1/2$,

$$\mathbf{P}\{X_i \geq 1/2\} = \left(\frac{d}{2}\right)\left(\frac{1}{4}-\left(\frac{1}{2}-c\right)^2\right) = \left(\frac{d}{2}\right)(c-c^2) = \frac{1-c}{2-c} ,$$

we have

$$\begin{aligned}
\mathbf{E}D_{n,i} &\geq \left(\frac{1-c}{2-c}\right)\lfloor m/2 \rfloor (1-\exp(-d(n-1)/(2m^2))) \\
&\geq \left(\frac{1-c}{2-c}\right)(1/2)\lfloor m/2 \rfloor \\
&\sim \frac{(1-c)\sqrt{dn}}{4(2-c)\sqrt{\log 4}}
\end{aligned}$$

if we choose $m = \lfloor \sqrt{d(n-1)/\log 4} \rfloor$. Note that as $c \to 0$, $d \to \infty$ and $(1-c)/(2-c) \to 1/2$. This implies that on average, the times required for SEARCH, INSERT, and DELETE grow as $\Theta(\sqrt{dn})$ for this family of distributions. The same is

true for the uniform distribution on the unit square rotated over an angle θ, with $\theta \in (0, \pi/2)$.

6. VERY DEPENDENT RANDOM PAIRS

Consider pairs (X, Y) so intertwined and dependent that the domination factor $d = \infty$. What can we say about H_n and H_n^*? It is futile to attempt to give the answer in each individual case: the behavior seems to be related to the dependence in a very intricate way. We are mainly interested in what can be said globally. One way of doing this is by giving upper bounds that are valid for large classes of distributions, but that may, in individual cases, be rather loose. Let us introduce the following dependence criterion: for $\alpha > 1$,

$$d_\alpha = \inf_{g,h} \left(\int \frac{f^\alpha(x, y)}{(g(x)h(y))^{\alpha-1}} \, dx \, dy \right)^{1/\alpha},$$

where g, h are arbitrary densities. We always have $d_\alpha \geq 1$ since by Jensen's inequality,

$$\int \frac{f^\alpha}{(gh)^{\alpha-1}} = \int \left(\frac{f}{gh}\right)^\alpha gh \geq \left(\int f\right)^\alpha = 1 .$$

We see that $d_\alpha = 1$ if and only if $f = gh$ almost everywhere, i.e., if and only if X and Y are independent. Next, we see that $d_\alpha \leq d$ in all cases because for any pair (g, h),

$$\left(\int \left(\frac{f}{gh}\right)^\alpha gh \right)^{1/\alpha} \leq \operatorname{ess\,sup} \frac{f}{gh} .$$

What makes d_α interesting is that it is finite for many more distributions: there are many examples in which $d = \infty$, yet $d_\alpha < \infty$ for some or all $\alpha > 1$ (see Example 11 below). In fact, there is much more structure in the problem: the collection $\{d_\alpha : \alpha > 1\}$ is increasing in α: indeed, for $\alpha < \beta$, and fixed f, g, h,

$$\left(\int \left(\frac{f}{gh}\right)^\alpha gh \right)^{1/\alpha} \leq \left(\int \left(\frac{f}{gh}\right)^\beta gh \right)^{1/\beta},$$

from which we conclude $d_\alpha \leq d_\beta$. As in the case of the domination factor, d_α somehow measures the degree of dependence between X and Y.

Example 8. To show that $d_\alpha < \infty$ for most densities of practical interest, just consider $\alpha = 2$. Then, taking g and h both as Cauchy densities $1/(\pi(1 + x^2))$,

$$\begin{aligned} d_2^2 &\leq \int \frac{f^2(x, y)}{g(x)h(y)} \, dx \, dy \\ &\leq \|f\|_\infty \mathbf{E}\{\pi^2(1 + X^2)(1 + Y^2)\} \\ &\leq \|f\|_\infty \pi^2 \sqrt{\mathbf{E}(1 + X^2)^2 \mathbf{E}(1 + Y^2)^2} , \end{aligned}$$

which is finite when f is bounded and both X and Y have finite fourth moments. In fact, d_2 is also finite if there exist monotone transformations of the axes that transform f into a density with these properties.

Example 9. By extending Example 8, it is easy to see that whenever f is bounded, and $\mathbf{E}|X|^\epsilon < \infty$ and $\mathbf{E}|Y|^\epsilon < \infty$ for some $\epsilon > 0$, we have $d_\alpha < \infty$ for some $\alpha > 1$.

Example 10. Assume that f has support on $[0,1]^2$. Then, taking g and h both uniform on $[0,1]$, we observe that $d_\alpha \le \int f^\alpha$. It should be noted that $\int f^\alpha$ is a measure of the peakedness of f. However, it may be finite even though f has infinitely many infinite peaks. The larger α, the more restrictive the peakedness condition.

We now proceed basically as in Theorem 1. The height is $O(n^{\alpha/(2\alpha-1)})$ in probability. The $O(\sqrt{n})$ height obtained in Theorem 1 is in a sense a limiting result as $\alpha \to \infty$. The main result is the following:

Theorem 2. *Consider a random Cartesian tree on n nodes under the general model. Assume that $d_\alpha < \infty$. Then, for every $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbf{P}\{H_n^* > (C+\epsilon)(d_\alpha n)^{\alpha/(2\alpha-1)}\} = 0\,,$$

where

$$C = 2e(4\log 4)^{(\alpha-1)/(2\alpha-1)}\,.$$

Also,

$$\limsup_{n\to\infty} \frac{\mathbf{E}\{H_n^*\}}{C(d_\alpha n)^{\alpha/(2\alpha-1)}} \le 1\,.$$

The same inequalities remain valid for H_n.

7. PROOF OF THEOREMS 1 AND 2

For Theorem 1, find densities g and h such that for every x, y,

$$f(x,y) \le \Delta g(x) h(y)\,,$$

where $\Delta = (d+\delta)$ and $\delta > 0$ is arbitrary. For Theorem 2, find densities g and h such that

$$\int \frac{f^\alpha(x,y)}{(g(x)h(y))^{\alpha-1}}\,dx\,dy \le \Delta^\alpha\,,$$

where $\Delta^\alpha = (d_\alpha^\alpha + \delta)$ and $\delta > 0$ is arbitrary. We consider a partition of the x-axis into intervals $A_1, \ldots, A_m$ such that $\int_{A_i} g(x)dx = 1/m$. Similarly, we find intervals

$B_1, \ldots, B_m$ such that $\int_{B_i} h(y)\,dy = 1/m$. We call a cell a Cartesian product of the form $A_i \times B_j$. Under the conditions of Theorem 1, observe that the probability of a cell is given by

$$\int_{A_i \times B_j} f(x, y)\,dx\,dy \le \Delta \int_{A_i} g(x)\,dx \times \int_{B_j} h(y)\,dy = \frac{\Delta}{m^2}.$$

Let S be a collection of cell indices of the type (i, j), and set

$$D = \bigcup_{(i,j)\in S} A_i \times B_j.$$

Under the conditions of Theorem 2, by Hölder's inequality,

$$\begin{aligned}\int_D f(x, y)\,dx\,dy &= \int_D \frac{f(x, y)}{g(x)h(y)}\, g(x)h(y)\,dx\,dy \\ &\le \left(\int_D \frac{f^\alpha(x, y)}{(g(x)h(y))^{\alpha-1}}\,dx\,dy\right)^{1/\alpha} \times \left(\int_D g(x)h(y)\,dx\,dy\right)^{1-1/\alpha} \\ &\le \Delta\left(\frac{|S|}{m^2}\right)^{1-1/\alpha}\end{aligned}$$

If $|S| = 2m$, then the number of points (N) in D is stochastically not greater than a binomial (n, p) random variable with $p = \Delta(2/m)^{1-1/\alpha}$, where formally $\alpha = \infty$ when the conditions of Theorem 1 are satisfied. By Chernoff's bounding method [7], for $t > 0$,

$$\begin{aligned}\mathbf{P}\{N > t\} &\le \mathbf{E}\{e^{\lambda(N-t)}\} \\ &\le \exp\left(t - np - t\log\left(\frac{t}{np}\right)\right)\end{aligned}$$

so that

$$\begin{aligned}\mathbf{P}\{N > (1+\epsilon)np\} &\le \exp(np(\epsilon - (1+\epsilon)\log(1+\epsilon))) \\ &= \exp(-np)\end{aligned}$$

when we take $\epsilon = e - 1$.

After these preliminaries, we can get on with the business of bounding H_n^*. We define a chain of cells as a sequence of cells with indices $(1, 1), \ldots, (m, m)$ having the property that if (i, j) is in the chain, then the next cell must have indices $(i+1, j)$ or $(i, j+1)$. The number of chains is

$$\binom{2m-2}{m-1} \le 2^{2m-2}.$$

Define an antichain of cells as a collection indexed by $(m, 1), \ldots, (1, m)$, with the restriction that (i, j) must be followed by $(i-1, j)$ or $(i, j+1)$. The number of chains equals the number of antichains. We let $N(\mathscr{C})$ denote the number of data points in the chain or antichain $\mathscr{C}$. The next claim is crucial:

$$H_n^* \le 1 + \max_{\text{all chains } \mathscr{C}} N(\mathscr{C}) + \max_{\text{all antichains } \mathscr{C}} N(\mathscr{C}) .$$

This uses the down-records, up-records argument of Devroye [17]: the depth of a node in a binary search tree is equal to the number of down-records observed in the subsequence of data points that have a larger key, plus up-records observed in the subsequence of data points that have a smaller key. Clearly, the nodes on the path formed by the down-records all belong to an antichain, while the up-records all belong to a chain. By Bonferroni's inequality, we have

$$\begin{aligned}\mathbf{P}\{H_n^* > 2t+1\} &\le 2\mathbf{P}\{\max_{\text{all chains } \mathscr{C}} N(\mathscr{C}) > t\} \\ &\le 2 \sum_{\text{all chains } \mathscr{C}} \mathbf{P}\{N(\mathscr{C}) > t\} \\ &\le 2\binom{2m-2}{m-1} e^{-np}\end{aligned}$$

when we take

$$t = enp = en\Delta(2/m)^{1-1/\alpha} .$$

Choose m in such a way that $m \to \infty$ with n. Then

$$\binom{2m-2}{m-1} \sim \frac{2^{2m-2}}{\sqrt{\pi m}}$$

so that for n large enough,

$$\mathbf{P}\{H_n^* > 2t+1\} \le \frac{2^{2m}}{\sqrt{\pi m}} \exp(-\Delta n(2/m)^{1-1/\alpha}) = o(1)$$

provided that

$$m = \left\lceil \left(\frac{\Delta n}{\log 4}\right)^{\alpha/(2\alpha-1)} 2^{(\alpha-1)/(2\alpha-1)} \right\rceil .$$

If $\alpha = \infty$, $\alpha/(2\alpha-1)$ should be replaced by its limit by continuous extension, i.e., $1/2$, and similarly for all other similar occurrences. With this choice,

$$t \ge e(n\Delta)^{\alpha/(2\alpha-1)} 2^{(2\alpha-2)/(2\alpha-1)} (\log 4)^{(\alpha-1)/(2\alpha-1)} .$$

Considering that Δ can be picked arbitrarily close to d or d_α, the first half of both theorems is proved. The second half follows simply from the bounds given above and $\mathbf{E}H_n^* = \int_0^\infty \mathbf{P}\{H_N^* > t\}\, dt$.

8. COUNTEREXAMPLES

It is useful to have a battery of distributions readily available for drawing counterexamples. Let us describe one such family, in which (X, Y) is distributed

on the top left triangle of $[0,1]\times[0,2]$. Assume that we have independent random variables W and U, where U is uniformly distributed on $[0,1]$, and W has a decreasing density φ on $[0,1]$. Its distribution function is denoted by Φ. Next, we define

$$(X, Y) = (U, U + W)$$

so that $Y - X = W$. The density of (X, Y) is given by

$$f(x, y) = \varphi(y - x) I_{0 \le x \le 1} .$$

For later reference, observe that

$$d_\alpha^\alpha \le 2^{\alpha-1} \int f^\alpha \le 2^\alpha \int \varphi^\alpha .$$

Partition $[0,1]\times[0,2]$ into a rectangular grid of m by $2m$ with sides equal to $1/m$. Mark the m grid cells that straddle the diagonal of $[0,1]^2$. Let $E_1, \ldots, E_m$ be the indicators of the events that the marked grid cells contain at least one data point, with E_1 referring to the cell with the smallest y-values, and so on up. A simple geometric argument shows that

$$H_n \ge L_n \ge \sum_{i=1}^{m} E_i ,$$

where L_n is the length of the left spine of the Cartesian tree. Hence, if the marked grid cells intersected with the upper left triangle yield the triangles $S_1, \ldots, S_m$,

$$\begin{aligned}
\mathbf{E}H_n &\ge \mathbf{E}L_n \ge m\mathbf{E}E_1 \\
&= m(1 - (1 - \mathbf{P}\{(X, Y) \in S_1\})^n) \\
&\ge m(1 - (1 - (1/(2m-1))\mathbf{P}\{W < 1/m\})^n) \\
&\ge m\left(1 - \exp\left(-\frac{n}{2m-1}\,\Phi(1/m)\right)\right) \\
&\ge m/2
\end{aligned}$$

if we choose m in such a way that

$$\frac{n\Phi(1/m)}{2m-1} \ge \log 2 .$$

It suffices that

$$R(m) \stackrel{\text{def}}{=} \frac{1}{m} \int_0^{1/m} \varphi(w)\, dw \ge \frac{\log 4}{n} .$$

We thus obtain

$$\mathbf{E}L_n \ge (1/2)\lfloor R^{\text{inv}}(\log 4/n)\rfloor .$$

If $\varphi(w)$ varies as $w^{-\alpha}$ as $w \to 0$ for $a \in [0, 1)$, then $\mathbf{E}L_n = \Omega(n^{1/(2-a)})$. The lower bound can thus grow at any polynomial rate between $\sqrt{n}$ and $o(n)$. When $a > 0$, it is easy to see that $d = \infty$, yet $d_\alpha < \infty$ for $\alpha < 1/a$. The upper bound of Theorem 2 implies that

$$\mathbf{E}H_n = O(n^{1/(2-a)+\epsilon})$$

for any $\epsilon > 0$. This shows that Theorem 2, in this instance, yields almost optimal results.

Example 11. The family described here can be used to show that there exist distributions with $d = \infty$, yet $d_\alpha < \infty$ for all α. Just take $\varphi(w) = -\log w$.

Example 12. If we take $\varphi(w) = c \max(1, 1/(w \log^2 w))$, where c is a normalization constant, then $R(m) \sim c/(m \log m)$ as $m \to \infty$, so that

$$\mathbf{E}L_n \geq \frac{(c + o(1))n}{\log 4 \log n} .$$

But the inequality of Theorem 2 then implies that $d_\alpha = \infty$ for all $\alpha > 1$.

9. RELATIONSHIP WITH OUTER LAYERS, MAXIMAL VECTORS

A maximal vector among the data is a pair $(X_i\ Y_i)$ such that no (X_j, Y_j) exists with both $X_j \geq X_i$ and $Y_j \geq Y_i$, $j \neq i$. The collection of maximal vectors forms the maximal layer. A common generalization of this is the outer layer, which consists of all data pairs (X_i, Y_i) such that one of the four quadrants centered at (X_i, Y_i) is empty. All maximal vectors are on the right spine of the Cartesian tree. If M_n is the number of maximal vectors, we have

$$M_n \leq H_n .$$

Thus, all the bounds of Theorems 1 and 2 apply as well to M_n. Let us collect these in Theorem 3, which generalizes results by the author [13, 16]:

Theorem 3. *Assume that $d_\alpha < \infty$. Then, for every $\epsilon > 0$,*

$$\lim_{n \to \infty} \mathbf{P}\{M_n > (C + \epsilon)(d_\alpha n)^{\alpha/(2\alpha-1)}\} = 0 ,$$

where

$$C = 2e(4 \log 4)^{(\alpha-1)/(2\alpha-1)} .$$

Also,

$$\limsup_{n \to \infty} \frac{\mathbf{E}\{M_n\}}{C(d_\alpha n)^{\alpha/(2\alpha-1)}} \leq 1 ,$$

and $\mathbf{E}M_n^r = O(n^{r\alpha/(2\alpha-1)})$ *for any* $r>0$. *If* d *is finite, then, for every* $\epsilon>0$,

$$\lim_{n\to\infty} \mathbf{P}\{M_n > (C+\epsilon)\sqrt{dn}\} = 0\,,$$

where $C = 4e\sqrt{\log 4}$. *Also,*

$$\limsup_{n\to\infty} \frac{\mathbf{E}\{M_n\}}{C\sqrt{dn}} \le 1\,,$$

and $\mathbf{E}M_n^r = O(n^{r/2})$ *for any* $r>0$.

Remark 4. If $d<\infty$, we have $\mathbf{E}M_n^r = O(n^{r/2})$ by Theorem 3. The same result could have been obtained by the moment inequalities of Devroye [12]. ∎

Remark 5: Finding the Maximal Vectors by Divide-and-Conquer. Two collections of maximal vectors can be merged in linear time if both are sorted according to the same coordinate. By the expected time analysis of divide-and-conquer algorithms given in [12] this means that the collection of maximal vectors can be found in linear expected time if

$$\sum_{n=1}^{\infty} \frac{\mathbf{E}M_n}{n^2} < \infty\,.$$

This result was rediscovered later by Clarkson and Shor [8, 9]. It is clear that under the conditions of Theorems 1 or 2, the divide-and-conquer algorithm described in [12] takes linear expected time. From Example 9, we recall that it suffices that the joint density f be bounded and that $\mathbf{E}|X|^\epsilon + \mathbf{E}|Y|^\epsilon < \infty$ for some $\epsilon>0$. This generalizes results in [5, 4, 11], and [13]. For other fast algorithms for maximal vectors, and some analysis, we refer to [18, 6, 21, 25, 3], and [27]. ∎

Remark 6: Convex Hull Algorithms. Assume that we find the convex hull by first finding the outer layer, which is known to contain the convex hull, and then applying a convex hull algorithm. Consider the time required for the second step only. With Graham's algorithm [23], we obtain an expected complexity equal to $O(\mathbf{E}M_n \log M_n) = O(\mathbf{E}M_n)\log n = o(n)$ when $d_\alpha < \infty$ for some $\alpha>0$ (see Example 9). Thus, on the average, finding convex hulls for these distributions is (complexity-wise) equivalent to finding outer layers. Interestingly, if $d<\infty$, and if a naive quadratic convex hull algorithm is used, the second step still takes linear expected time because $\mathbf{E}M_n^2 = O(n)$ (see Theorem 3). It should be stressed that the linear expected time algorithms for convex hulls obtained in this manner do not rely on hashing or truncation operations: the linearity is simply the result of the sparseness of the sought objects, in this case, the outer layer and the convex hull. ∎

10. CONCLUSION

The big disappointment with Cartesian trees is that standard dictionary and priority queue operations take about $\sqrt{n}$ instead of $\log n$ time on the average

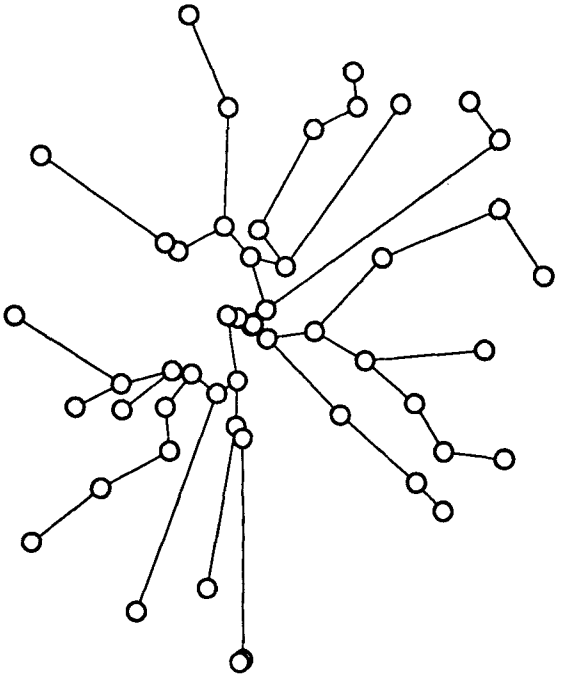

Fig. 5. Random polar Cartesian tree, in which points are placed at their true x and y coordinates.

when we work with real data. Only the artificial case in which X and Y are independent offers a $\log n$ behavior. This suggests the following exciting prospect: assume that both X and Y are $[0, 1]$-valued. Why don't we add artificially generated points with the uniform distribution on the unit square? If the original distribution has a bounded density on $[0, 1]^2$, then adding n such points yields a Cartesian tree with $2n$ points and $\Theta(\log n)$ expected time per standard operation. Thus, adding points reduces the expected time! This paradigm certainly deserves more attention.

For locating points in the plane in computational geometric applications, one might consider a polar Cartesian tree, which is a Cartesian tree based upon (Θ, R) pairs, where $R^2 = X^2 + Y^2$ and $\tan \Theta = Y/X$ as in the standard polar transformation. Such a tree is shown in Fig. 5.

ACKNOWLEDGEMENT

I would like to thank both referees.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[2] C. R. Aragon and R. G. Seidel, Randomized search trees, in *Proceedings of the 29th IEEE Symposium on the Foundations of Computer Science*, 1988, pp. 1–6.

[3] J. L. Bentley, K. L. Clarkson, and D. B. Levine, Fast linear expected-time algorithms for computing maxima and convex hulls, in *Proceedings of the First Annual ACM–SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, PA, 1990, pp. 179–187.

[4] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson, On the average number of maxima in a set of vectors and applications, *J. Assoc. Comput. Mach.*, **25**, 536–543 (1982).

[5] J. L. Bentley and M. I. Shamos, Divide and conquer for linear expected time, *Inf. Process. Lett.*, **7**, 87–91 (1978).

[6] C. Buchta, On the average number of maxima in a set of vectors, *Inf., Process. Lett.*, **33**, 63–65 (1989).

[7] H. Chernoff, A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations, *Ann. Math. Stat.* **23**, 493–507 (1952).

[8] K. L. Clarkson and P. W. Shor, Algorithms for diametrical pairs and convex hulls that are optimal, randomized, and incremental, in *Proceedings of the Fourth Symposium on Computational Geometry, ACM*, New York, 1988, pp. 12–17.

[9] K. L. Clarkson and P. W. Shor, Applications of random sampling in computational geometry II, *Discrete Comput. Geom.*, **4**, 387–422 (1989).

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[11] L. Devroye, A note on finding convex hulls via maximal vectors, *Inf., Process. Lett.*, **11**, 53–56 (1980).

[12] L. Devroye, Moment inequalities for random variables in computational geometry, *Computing*, **30**, 111–119 (1983).

[13] L. Devroye, On the expected time required to construct the outer layer, *Inf., Proc. Lett.*, **20**, 255–257 (1985).

[14] L. Devroye, A note on the height of binary search trees, *J. Assoc. Comput. Mach.*, **33**, 489–498 (1986).

[15] L. Devroye, *Lecture Notes on Bucket Algorithms*, Birkhäuser Verlag, Boston, MA, 1986.

[16] L. Devroye, Branching processes in the analysis of the heights of trees, *Acta Inf.*, **24**, 277–298 (1987).

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

[18] R. A. Dwyer, Kinder, gentler average-case analysis for convex hulls and maximal vectors, *SIGACT News*, **21**, 64–71 (1990).

[19] P. Flajolet and A. Odlyzko, The average height of binary trees and other simple trees, *J. Comput. Syst. Sci.*, **25**, 171–213 (1982).

[20] J. Françon, G. Viennot, and J. Vuillemin, Description and analysis of an efficient priority queue representation, in *Proceedings of the 19th IEEE Symposium on the Foundations of Computer Science*, 1978, pp. 1–7.

[21] H. N. Gabow, J. L. Bentley, and R. E. Tarjan, Scaling and related techniques for geometry problems, in *Proceedings of the 16th Annual Symposium on the Theory of Computing*, 1984, pp. 135–143.

[22] G. H. Gonnet, *A Handbook of Algorithms and Data Structures*, Addison-Wesley, Reading, MA, 1984.

[23] R. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inf., Proc. Lett.*, **1**, 132–133 (1972).

[24] R. Kemp, *Fundamentals of the Average Case Analysis of Particular Algorithms*, B. G. Teubner, Stuttgart, Germany, 1984.

[25] D. G. Kirkpatrick and R. Seidel, Output-sensitive algorithms for finding maximal

vectors, in *Proceedings of the 2nd Annual Symposium on Computatational Geometry, Baltimore*, ACM, New York, 1985, pp. 89–96.
[26] D. E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.
[27] H. T. Kung, F. Luccio, and F. P. Preparata, On finding the maxima of a set of vectors, *J. Assoc. Comput. Mach.*, **22**, 469–476 (1975).
[28] G. Louchard, Exact and asymptotic distributions in digital and binary search trees, *Theor. Inf., Appl.*, **21**, 479–496 (1987).
[29] W. C. Lynch, More combinatorial problems on certain trees, *Comput. J.*, **7**, 299–302 (1965).
[30] H. M. Mahmoud, On the average internal path length of m-ary search trees, *Acta Inf.*, **23**, 111–117 (1986).
[31] H. M. Mahmoud, *Evolution of Random Search Trees*, Wiley, New York, 1992.
[32] H. M. Mahmoud and B. Pittel, On the joint distribution of the insertion path length and the number of comparisons in search trees, *Discrete Appl. Math.*, **20**, 243–251 (1988).
[33] H. Mahmoud and B. Pittel, On the most probable shape of a search tree grown from a random permutation, *SIAM J. Algebraic Discrete Methods*, **5**, 69–81 (1984).
[34] E. M. McCreight, Priority search trees, *SIAM J. Comput.*, **14**, 257–276 (1985).
[35] B. Pittel, On growing random binary trees, *J. Math. Anal. Appl.* **103**, 461–480 (1984).
[36] J. M. Robson, The height of binary search trees, *Aust. Comput. J.*, **11**, 151–153 (1979).
[37] R. Sedgewick, Mathematical analysis of combinatorial algorithms, in *Probability Theory and Computer Science*, G. Louchard and G. Latouche, Eds., Academic Press, London, England, 1983, pp. 123–205.
[38] R. E. Tarjan, *Data Structures and Network Algorithms*, CBMS 44, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.
[39] J. S. Vitter and P. Flajolet, Average-case analysis of algorithms and data structures, in *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, J. van Leeuwen, Ed., MIT Press, Amsterdam, The Netherlands, 1990, pp. 431–524.
[40] J. Vuillemin, A data structure for manipulating priority queues, *Commun. ACM*, **21**, 309–314 (1978).
[41] J. Vuillemin, A unifying look at data structures, *Commun. ACM*, **23**, 229–239 (1980).
[42] R. L. Wheeden and A. Zygmund, *Measure and Integral*, Marcel Dekker, New York, 1977.

Received January 13, 1992
Revised January 27, 1993