DESCENDANTS IN HEAP ORDERED TREES OR A TRIUMPH OF COMPUTER ALGEBRA

HELMUT PRODINGER

Institut für Algebra und Diskrete Mathematik
Technical University of Vienna
Wiedner Hauptstrasse 8–10
A-1040 Vienna, Austria.
email: `Helmut.Prodinger@tuwien.ac.at`
www: `http://info.tuwien.ac.at/theoinf/proding.htm`

Submitted: June 8, 1996; Accepted: September 16, 1996.

I dedicate this paper to Doron Zeilberger and his program EKHAD.

ABSTRACT. A heap ordered tree with n nodes ("size n") is a planted plane tree together with a bijection from the nodes to the set $\{1, \ldots, n\}$ which is monotonically increasing when going from the root to the leaves. We consider the number of descendants of the node j in a (random) heap ordered tree of size $n \geq j$. Precise expressions are derived for the probability distribution and all (factorial) moments.

AMS Subject Classification. 05A15 (primary) 05C05 (secondary)

1. HEAP ORDERED TREES

A *heap ordered tree* with n nodes ("size n") might be described as a *planted plane tree* together with a bijection from the nodes to the set $\{1, \ldots, n\}$ which is *monotonically increasing* when going from the root to the leaves.

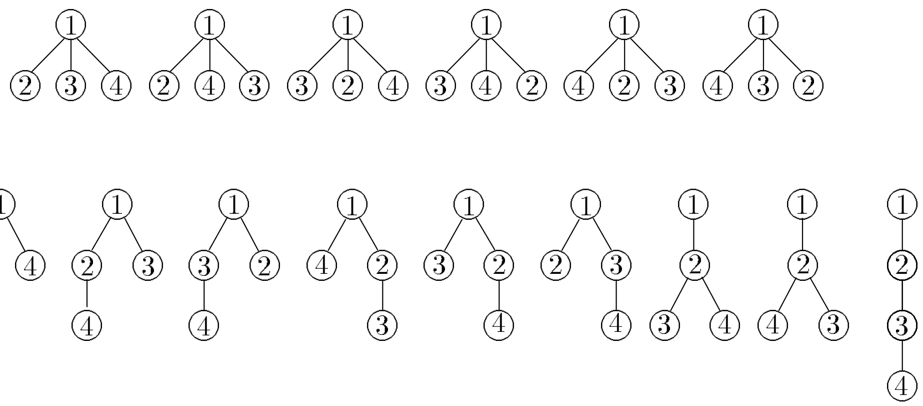

FIGURE 1. All 15 heap ordered trees with 4 nodes

In this note, we want to concentrate on the **number of descendants** of the node j in a (random) *heap ordered tree* of size $n \geq j$. By convention, we say that the node j is a descendant of itself. For instance, node 1 always has n descendants and node n has 1 descendant.

The interest in the number of descendants stems from the Ph. D. thesis of Janice Lent [4]; compare also [5]. In [6], Conrado Martínez and myself are currently investigating this parameter (and several others) for binary search trees and some variants.

For more information about heap ordered trees the reader is referred to [1], [9], [8].

We will get explicit formulæ for all factorial moments, as well as for the probability generating functions. Finally, even the probabilities themselves can be computed explicitly.

Methodologically, we first got the results for the moments by guessing with MAPLE. Then the proofs of the obtained formulæ are done mechanically with Zeilberger's algorithm EKHAD, compare [2] and [7]. We assume that the reader is familiar with this very important new feature.

The enumeration of the numbers a_n of heap ordered trees of size n is easy and appears already in [10]. The recursion is

$$a_{n+1} = \sum_{m\geq 1} \sum_{h_1+\cdots+h_m=n} \binom{n}{h_1,\ldots,h_m} a_{h_1}\ldots a_{h_m} \quad \text{for } n \geq 1, \quad a_1 = 1\ ; \tag{1}$$

hence the exponential generating function $A(z) := \sum_{n\geq 0} a_n \frac{z^n}{n!}$ fulfills the differential equation

$$A'(z) = \frac{1}{1-A(z)} \qquad \text{with} \quad A(0) = 0\ ,$$

with the solution

$$A(z) = 1 - \sqrt{1-2z}\ ,$$

so that

$$a_n = n!\, 2^{1-n}\, \mathcal{C}_n = 1\cdot 3\cdot 5\ldots(2n-3)\ ,$$

with a (shifted) *Catalan number* $\mathcal{C}_n = \frac{1}{n}\binom{2n-2}{n-1}$.

This formula can be extended to the complex plane by means of GAMMA functions. Then it turns out that $a_0 = -1$ is the *natural* value. We will work with this redefined value in the sequel.

We want the probability that j lies in a subtree of size k. For that purpose we first note the alternative recursion

$$a_{n+1} = \sum_{m\geq 1} \sum_{h_1+\cdots+h_m=n} m \binom{n-1}{h_1-1, h_2, \ldots, h_m} a_{h_1}\ldots a_{h_m} \quad \text{for } n \geq 1, \quad a_1 = 1\ .$$

It is obtained by forcing the node j to be in the first subtree, thus restricting the generality, which is restored by introducting a factor m. From this we get the desired probability as

$$\sum_{m\geq 1} \sum_{h_2+\cdots+h_m=n-k} m \binom{n-1}{k-1} \binom{n-k}{h_2,\ldots,h_m} \frac{a_k a_{h_2}\ldots a_{h_m}}{a_{n+1}}\ .$$

We can pull out the factor $\binom{n-1}{k-1}\frac{a_k}{a_{n+1}}$; the exponential generating function of the remaining sum is amazingly simple:

$$\frac{d}{du}\frac{u}{1-u\,A(z)}\bigg|_{u=1} = \frac{1}{1-2z}\ ,$$

whence our sought probability that j lies in a subtree of size k turns out to be

$$\binom{n-1}{k-1}\frac{a_k}{a_{n+1}}\,2^{n-k}\,(n-k)!\ .$$

2. Results

Now, let $F_{n,j}(v)$ be the probability generating function of the number of descendants, i.e. the coefficient of v^m in $F_{n,j}(v)$ is the probability that node j in a random heap ordered tree with n nodes has m descendants.

We must take care of the fact that in its subtree, 'j' does not mean 'j' any further. The numbers in the subtree are to be replaced by $1, 2, \ldots$, according to their relative order. Let us compute the probability that j will be i after this procedure, or, what is the same, that j is the ith largest number in its subtree. It is

$$\frac{\binom{j-2}{i-1}\binom{n+1-j}{k-i}}{\binom{n-1}{k-1}}\ ,$$

since $i-1$ numbers have to be chosen from $\{2, 3, \ldots, j-1\}$ and $k-i$ numbers from $\{j+1, \ldots, n+1\}$.

Hence we get our recursion for the probability generating functions.

$$F_{n+1,j}(v) = \sum_{k=1}^{n}\binom{n-1}{k-1}\frac{a_k}{a_{n+1}}\,2^{n-k}\,(n-k)!\sum_{i=1}^{k}\frac{\binom{j-2}{i-1}\binom{n+1-j}{k-i}}{\binom{n-1}{k-1}}\,F_{k,i}(v)\ . \tag{2}$$

This holds for $n \geq 1$ and $j \geq 2$; the initial conditions are $F_{n,1}(v) = v^n$. The recursion should be self-explanatory.

After one sunday afternoon with Maple, I found this explicit form of the probability generating functions (by some sort of creative guessing).

Theorem 1. *The probability generating function of the parameter* number of descendants *of node j in a random heap ordered tree of size n is given by*

$$F_{n,j}(v) = \sum_{s=0}^{n+1-j}\frac{a_s\,a_j}{a_{s+j}}\Big((2s-1)n+j-1\Big)\,(n-j)^{\underline{s-1}}\,\frac{(v-1)^s}{s!}\ , \tag{3}$$

where $x^{\underline{n}} = x(x-1)\ldots(x-n+1)$, which is the notation for the falling factorials from [2].

Before we go to the proof, we get expectation and variance as a corollary. The expectation is the coefficient of $(v-1)$ in $F_{n,j}$, i.e.

$$\frac{n+j-1}{2j-1}\ .$$

The second factorial moment is twice the coefficient of $(v-1)^2$ in $F_{n,j}$, i.e.

$$\frac{a_2\,a_j}{a_{2+j}}(3n+j-1)\,(n-j)^{\underline{1}} = \frac{(3n+j-1)(n-j)}{(2j+1)(2j-1)}\ .$$

Theorem 2. *The expectation and the variance of the parameter* number of descendants *of node j in a random heap ordered tree of size n is given by*

$$\mathsf{Expectation} = \frac{n+j-1}{2j-1}\ ,$$
$$\mathsf{Variance} = \frac{2(j-1)(2n-1)(n-j)}{(2j+1)(2j-1)^2}\ .$$

First a quick check that the announced formula is true for $j=1$:

$$\begin{aligned}F_{n,1}(v) &= \sum_{s\ge 0}\frac{a_s}{a_{s+1}}\,(2s-1)n\,(n-1)^{\underline{s-1}}\,\frac{(v-1)^s}{s!}\\ &= \sum_{s\ge 0} n^{\underline{s}}\,\frac{(v-1)^s}{s!}\\ &= \sum_{s\ge 0}\binom{n}{s}(v-1)^s = v^n\ .\end{aligned}$$

Now we assume $j\ge 2$, so that the recursion formula (2) is applicable. If we compare coefficients, we have to prove the following:

$$\begin{aligned}&\frac{a_s\,a_j}{a_{s+j}}\Big((2s-1)(n+1)+j-1\Big)\,(n+1-j)^{\underline{s-1}} =\\ &= \sum_{k=1}^{n}\frac{a_k}{a_{n+1}}\,2^{n-k}\,(n-k)!\sum_{i=1}^{k}\binom{j-2}{i-1}\binom{n+1-j}{k-i}\frac{a_s\,a_i}{a_{s+i}}\Big((2s-1)k+i-1\Big)\,(k-i)^{\underline{s-1}}\ .\end{aligned} \tag{4}$$

This looks definitely frightening, but in order to go on it is natural to interchange the summation on the right hand side, viz.

$$\frac{a_s}{a_{n+1}}\sum_{i=1}^{n}\binom{j-2}{i-1}\frac{a_i}{a_{s+i}}\sum_{k=i}^{n}a_k\,2^{n-k}\,(n-k)!\binom{n+1-j}{k-i}\Big((2s-1)k+i-1\Big)\,(k-i)^{\underline{s-1}}\ . \tag{5}$$

In the next section we will prove the *combinatorial identity* (4).

3. Proof of identity (4)

Lemma 1.

$$\begin{aligned}&\sum_{k=i}^{n}a_k\,2^{n-k}\,(n-k)!\binom{n+1-j}{k-i}\Big((2s-1)k+i-1\Big)\,(k-i)^{\underline{s-1}}\\ &= \frac{2^{2j-2i-n-1}\big((n+1)(2s-1)+j-1\big)(2n)!(j-i-1)!(2i+2s-3)!(n+1-j)!(j+s-2)!}{n!(i+s-2)!(n+2-j-s)!(2j+2s-3)!}\ .\end{aligned}$$

Proof. As announced earlier, we use Zeilberger's algorithm. We might note that the actual range of summation is $i+s-1 \le k \le n+1+i-j$. This sum, if given to EKHAD, produces a first order recursion, and is therefore expressible in closed form. $\square$

Remark. The sum can be separated naturally into two sums, according to

$$\Big((2s-1)k+i-1\Big) = ((2s-1)k) + (i-1) \ .$$

Both sums are also expressible in closed form. (When I prepared the first draft of this paper, I used an older version of EKHAD that produced only a second order recursion, so my strategy was to study the two sums separately.)

With this lemma, the inner sum of (5) has been evaluated, and we can turn to the whole sum (5). Define

$$F(n,i) := \binom{j-2}{i-1} \frac{a_i}{a_{s+i}} \times$$

$$\times \frac{2^{2j-2i-n-1}((n+1)(2s-1)+j-1)(2n)!(j-i-1)!(2i+2s-3)!(n+1-j)!(j+s-2)!}{n!(i+s-2)!(n+2-j-s)!(2j+2s-3)!} \ .$$

Zeilberger's algorithm shows that it is *Gosper summable*, or, to be concrete, if

$$G(n,i) := 2(i-1)\,F(n,i)$$

then

$$F(n,i) = G(n,i+1) - G(n,i) \ .$$

The range of summation is actually from $i=1$ to $i=j-1$, so our desired sum is $\frac{a_s}{a_{n+1}} G(n,j)$.

Finally, we ask MAPLE to simplify this minus the predicted result;

$$\frac{a_s}{a_{n+1}} G(n,j) - \frac{a_s\, a_j}{a_{s+j}} \Big((2s-1)(n+1)+j-1\Big)\, (n+1-j)^{\underline{s-1}}$$

and we get zero, so that the proof is finished.

4. Explicit probabilities

Now we can even get an explicit expression for the probability $p_{n,j;m}$ that node j in a random heap ordered tree of size n has m descendants, since this quantity is given by

$$[v^m]F_{n,j}(v) = \sum_{s=m}^{n+1-j} \frac{a_s\, a_j}{a_{s+j}} \Big((2s-1)n+j-1\Big)\, (n-j)^{\underline{s-1}} \frac{\binom{s}{m}(-1)^{s-m}}{s!} \ .$$

Giving this sum to Zeilberger's algorithm we see that we get a recursion of order one. Consequently, the sum is of closed form (or rather: can be brought into); the result is the following.

Theorem 3. *The probability $p_{n,j;m}$ that node $j \ge 2$ in a random heap ordered tree of size n has m descendants is given by*

$$p_{n,j;m} = \frac{4^{n-m+1-j}\,(n-j)!\,(2m-2)!\,(2j-2)!\,(n-m-1)!\,(n-1)!}{(m-1)!^2\,(2n-2)!\,(j-1)!\,(j-2)!\,(n-j-m+1)!} \ .$$

For $j = 1$ we have

$$p_{n,1;m} = \delta_{n,m} \ .$$

The extra case $j = 1$ can be considered to be a limiting case, since

$$\lim_{\epsilon \to 0} p_{n,1+\epsilon;m-\epsilon} = \delta_{n,m} \ .$$

The fact that probabilities sum to 1 leads to the identity

$$\sum_{0<m<n} 4^{-m} \binom{2m-2}{m-1} \binom{n-1-m}{j-2} = 4^{-n-1+j} \binom{2n-2}{n-1} \binom{n-1}{j-1} \binom{2j-2}{j-1}^{-1} \ .$$

This, too, is easy to prove directly by Zeilberger's algorithm.

5. Combinatorial derivation of the probabilities

The existence of the relatively simple formula for the probabilities as in Theorem 3 makes us optimistic to find a direct (combinatorial) proof for it. Here it is:

First, the formula

$$a_{n+1} = a_n \, (2n-1) \ ,$$

can be seen like this. From each tree with n nodes, we get $2n-1$ new trees by inserting the new node $n+1$. We can attach this new node to every node, but the relative order in the plane is important, so if a certain node has i outgoing branches, it gives us $i+1$ possibilities, namely to the left of all, between first and second edge, etc. Denoting by $d(k)$ the number of outgoing branches of node k, we have altogether

$$\sum_{k=1}^{n} \Big(1 + d(k)\Big) = n + \text{number of edges} = 2n-1 \ ,$$

as desired.

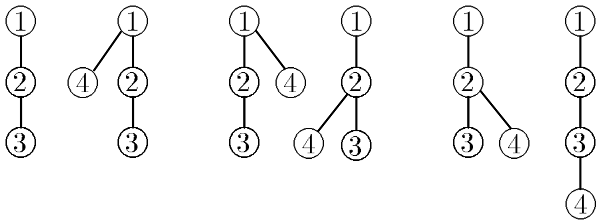

Figure 2. A heap ordered tree with 3 nodes and the 5 trees obtained by inserting a fourth node

Now we can use this idea to count the number of descendants of node j. We start with a (fixed) tree of size j and throw in new nodes at random. For instance, there is a probability $\frac{1}{2j-1}$ that the subtree with root j "catches" node $j+1$. It is a little bit like the cookie monster, described in [3]: the larger the monster is already, the likelier will it catch the next thrown cookie. We have collected the appropriate transition probabilities in a diagram (Figure 3) that resembles Pascal's triangle. Each node has two entries, the first one being the current total size, and the second one being the size of the subtree

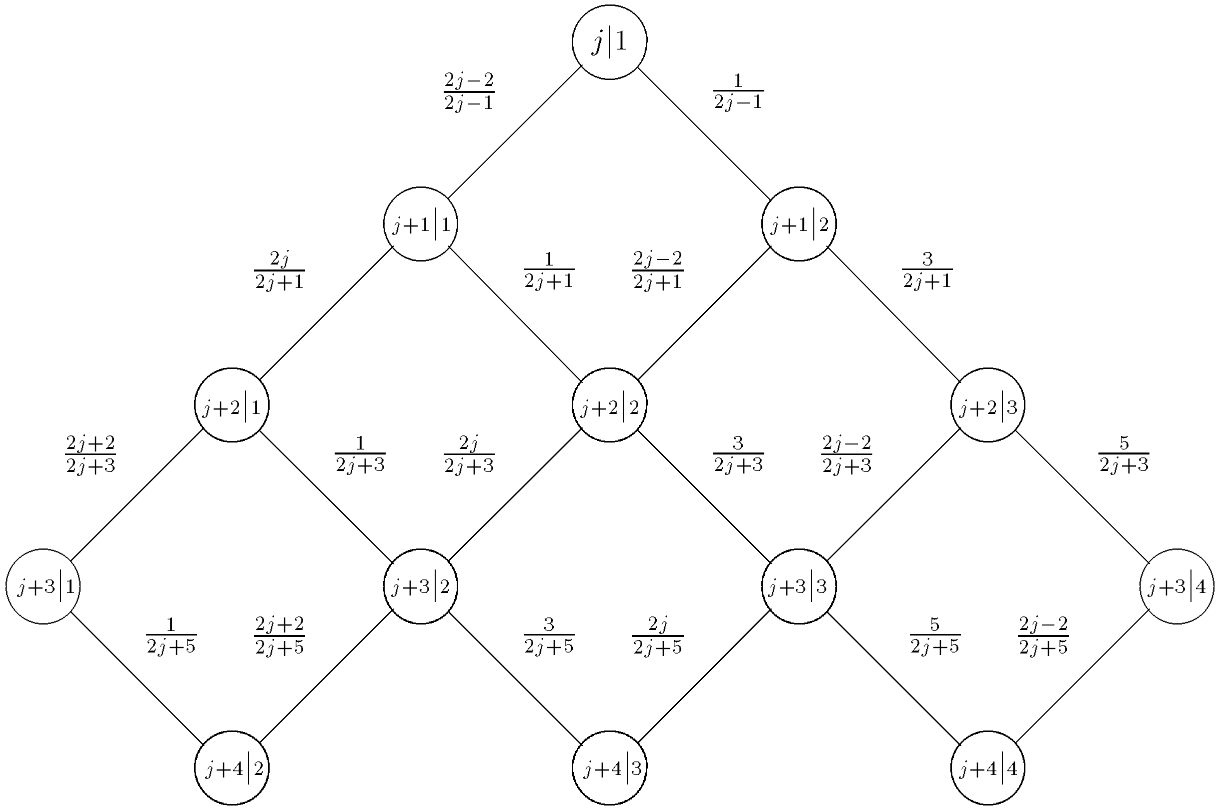

Figure 3. The beginning of the Pascal like grid

with root j. We start in state $j|1$ and can go left or right in each step. We are interested in the weight of all the paths leading to node $n|m$, since it means simply the probability that a random tree of size n has a subtree rooted at j of size m. That we start with a fixed tree does not matter, as can be easily seen. Now, regardless as we walk, we will "collect" a denominator

$$(2j-1)\cdot(2j+1)\ldots(2n-3)=\frac{a_n}{a_j}$$

and a numerator

$$(2j-2)\cdot(2j)\ldots(2n-2m-2)\;\cdot\;1\cdot3\ldots(2m-3)=\frac{(n-m-1)!\,2^{n-m-j+1}}{(j-2)!}\,a_m\ .$$

Having taken care of that factor, we only have to *count* the number of paths in question. It is easy to see that we get $\binom{n-j}{m-1}$ paths from state $j|1$ to state $n|m$. Altogether we find

$$p_{n,j;m}=\frac{(n-m-1)!\,2^{n-m-j+1}}{(j-2)!}\binom{n-j}{m-1}\frac{a_m\,a_j}{a_n}\ .$$

It is easy to see that this formula is equivalent to the previous one.

6. Binary trees

For the sake of completeness and comparison we briefly sketch the corresponding considerations for the instance of binary trees. They are considerably easier and we only list the results.

The recursion for the increasing binary trees is

$$a_{n+1} = \sum_{k=0}^{n} \binom{n}{k} a_k a_{n-k}$$

with the obvious solution $a_n = n!$. The probability that j lies in a subtree of size k is

$$2\binom{n-1}{k-1} \frac{k!(n-k)!}{(n+1)!} = \frac{2k}{n(n+1)} \; .$$

The recursion for the probability generating functions is

$$F_{n+1,j}(v) = \sum_{k=1}^{n} \frac{2k}{n(n+1)} \sum_{i=1}^{k} \frac{\binom{j-2}{i-1}\binom{n+1-j}{k-i}}{\binom{n-1}{k-1}} F_{k,i}(v) \; .$$

The solution is

$$F_{n,j}(v) = \sum_{s=0}^{n+1-j} \frac{s!\, j!}{(s+j)!} \Big((s+1)n - (j-1) \Big) \, (n-j)^{\underline{s-1}} \frac{(v-1)^s}{s!} \; .$$

Expectation and variance are given by

$$\mathsf{Expectation} = \frac{2n-(j-1)}{j+1} \; ,$$

$$\mathsf{Variance} = \frac{2(j-1)(n+1)(n-j)}{(j+2)(j+1)^2} \; .$$

The probabilities are given by

$$p_{n,j;m} = j(j-1)m \frac{(n-m-1)!(n-j)!}{n!(n-j-m+1)!} \; .$$

Acknowledgement. The insightful comments of an anonymous referee are gratefully acknowledged.

References

[1] W.-C. Chen and W.-C. Ni. On the average altitude of heap–ordered trees. *International Journal of Foundations of Computer Science*, 15:99–109, 1994.

[2] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics (Second Edition)*. Addison Wesley, 1994.

[3] D. H. Greene and D. E. Knuth. *Mathematics for the analysis of algorithms.* Birkhauser, Boston, second edition, 1982.

[4] J. Lent. *Probabilistic Analysis of some searching and sorting algorithms.* PhD thesis, George Washington University, 1996.

[5] J. Lent and H. M. Mahmoud. Average–case analysis of multiple quickselect: An algorithm for finding order statistics. *Statistics and Probability Letters*, 28:299–310, 1996.

[6] C. Martínez and H. Prodinger. On the number of descendants and ascendants in random search trees. *In preparation*, 1996.

[7] M. Petkovsek, H. Wilf, and D. Zeilberger. $A = B$. A.K. Peters, Ltd., 1996.
[8] H. Prodinger. The level of nodes in heap ordered trees. *submitted*, 1996.
[9] H. Prodinger. Depth and path length of heap ordered trees. *International Journal of Foundations of Computer Science*, 1996 (to appear).
[10] H. Prodinger and F.J. Urbanek. On monotone functions of tree structures. *Discrete Applied Mathematics*, 5:223–239, 1983.