Counting Balanced Tree Shapes

Jeffrey A. Barnett*

Keywords: Balanced trees, tree shapes, combinatorial problems, data structures.

1 Preliminaries

Tree structures are used to organize data for efficient retrieval. Some restrictions on possible shapes are usually imposed to guarantee that efficiency. One typical restriction is that sibling subtrees must have approximately the same height. Another is that sibling subtrees must have approximately the same size measured in nodes. Both restrictions, supplemented by suitable node labeling schemes, lead to $O(\log z)$ retrieval times, where z is the number of nodes in the tree. Such trees are said to be *balanced*. AVL trees [5], B-trees [2], and Red-Black trees [3] are some well-known examples of balanced trees. Knuth [4] provides algorithms and analyses for constructing, searching, and maintaining balanced trees of various sorts.

This article develops formulas that count the number of shapes of trees tightly balanced by subtree size. These trees will have a constant branching factor or width, w. So a (sub) tree is either "nothing" or a root node with w subtrees. A node is balanced if the difference in size of each pair of its subtrees is at most one node. A tree is balanced if all of its nodes are balanced. A size-balanced tree is also height-balanced in the sense that all pairs of root-to-leaf paths differ in length by one at most.

Here is the definition of '$\doteq$' the same shape predicate: 1) $\lambda \doteq \lambda$, where λ is the tree with no nodes; 2) if $u_i \doteq v_i$, where $1 \leq i \leq w$, u is a node with the ordered list of subtrees u_i, and v is a node with the ordered list of subtrees v_i, then $u \doteq v$. In other words, two shapes are the same if they look the same when drawn on paper. If $u \doteq v$, then u and v necessarily have the same number of nodes.

*The author's email is jbb@notatt.com.

2 Basic Formulas

Let $s_w(z)$, where $z \geq 0$ and $w \geq 1$, be the number of shapes of balanced trees with branching factor w and z nodes. Recurrence formulas used to calculate s_w are developed in this section. These formulas relate various values of $s_w(z)$ to $s_{w'}(z')$ where $w = w'$. In other words, there is a set of formulas for each $w > 1$. It is easy to see that $s_1(z) = 1$ for all $z \geq 0$. Henceforth, assume $w > 1$. Clearly $s_w(0) = 1$ because the empty tree shape is unique and $s_w(1) = 1$ because the only tree shape with a one node is the one with a root and w empty children.

The recurrence relation for larger tree sizes is

$$s_w(nw + 1 + m) = \binom{w}{m} s_w(n+1)^m s_w(n)^{w-m}, \quad \text{where } 0 \leq m \leq w \text{ and } n \geq 0. \tag{1}$$

If a tree has $nw+1+m$ nodes, one node must be the root, m children of the root must have $n+1$ nodes, and the remaining $w-m$ children must have n nodes; the tree can be balanced in no other way. Now observe that 1) the m of w subtrees with $n+1$ nodes can be chosen in $\binom{w}{m}$ ways and 2) the choice of the shape of each subtree is independent of the others. So formula (1) follows. The case where $n = 0$ is of special interest:

$$s_w(1 + m) = \binom{w}{m} \qquad \text{where } 0 \leq m \leq w. \tag{2}$$

Sequences $s_2, \ldots, s_7$ are registered on the Bell Labs Sequence Server [1].

3 Basic Properties

Several basic properties of $s_w(z)$ are observed and justified in this section.

Theorem 1. *$s_w(nw + 1 + 0) + \cdots + s_w(nw + 1 + w) = x^w$ for some integer x.*

Proof. Simply substitute from (1)

$$\begin{aligned} \sum_{m=0}^{w} s_w(nw + 1 + m) &= \sum_{m=0}^{w} \binom{w}{m} s_w(n+1)^m s_w(n)^{w-m} \\ &= \big(s_w(n+1) + s_w(n)\big)^w. \end{aligned}$$

□

Theorem 2. *$s_w(z) = \binom{w}{0}^{x_0} \binom{w}{1}^{x_1} \binom{w}{2}^{x_2} \cdots \binom{w}{w}^{x_w}$ for some nonnegative integers $x_0, \ldots, x_w$.*

Proof. $s_w(0) = s_w(1) = 1$ so the claim is true if $z = 0$ or 1. Values of $s_w(z)$, when $z = nw + 1 + m$ and $n \geq 0$ are, by (1), products of suitable binomial coefficients and values of s_w that are suitable by induction. □

Corollary 3. *All values of $s_2(z)$ are integer powers of 2. All values of $s_3(z)$ are integer powers of 3. For all $w > 3$, there are z such that $s_w(z)$ is not an integer power of w.*

Proof. Since the values of the binomial coefficients used to generate s_2 are 1, 2, and 1, the result for $w = 2$ follows. Similarly, the values of the binomial coefficients used to generate s_3 are 1, 3, 3, and 1, so the result follows for $w = 3$. When $w > 3$, some $\binom{w}{m}$ for $0 \leq m \leq w$ will not be a power of w, e.g., $s_w(3) = \binom{w}{2} = w(w-1)/2$ is not a power of w. □

Let σ_w^n be the value of the base-w number consisting of $n + 1$ 1 digits, i.e.,

$$\sigma_w^n = \sum_{i=0}^{n} w^i.$$

N.B. $\sigma_w^{n+1} = w \cdot \sigma_w^n + 1$.

Theorem 4. *$s_w(\sigma_w^n) = 1$ for all $n \geq 0$.*

Proof. The claim is certainly true when $n = 0$ because $s_w(\sigma_w^0) = s_w(1) = 1$. Now assume $s_w(\sigma_w^n) = 1$ for all $n \leq x$. Then

$$\begin{aligned} s_w(\sigma_w^{x+1}) &= s_w(w\sigma_w^x + 1) \\ &= \binom{w}{0} s_w(\sigma_w^x + 1)^0 s_w(\sigma_w^x)^w \\ &= 1 \end{aligned}$$

using (1) and the inductive assumption. □

The unique tree shape with σ_w^n nodes is the one where every root-to-leaf path is length n, i.e., the tree is perfectly balanced and the w children of each leaf—there are w^{n+1} in total—are all λ. As nodes are added, some of the λ are replaced with new leaves. When w^{n+1} nodes have been added, the tree is again perfectly balanced with σ_w^{n+1} nodes and all root-to-leaf paths are now length $n + 1$.

Theorem 5. *The sequence* $s_w(\sigma_w^n), s_w(\sigma_w^n+1), \ldots, s_w(\sigma_w^{n+1})$ *is symmetric in the sense that* $s_w(\sigma_w^n + z) = s_w(\sigma_w^{n+1} - z)$ *for all* $0 \le z \le w^{n+1}$.

Proof. If $z = 0$ or $z = w^{n+1}$ the claim follows from Theorem 4. When $0 < z < w^{n+1}$, let $z = \sum_{i=1}^{n} z_i w^i$, where $0 \le z_i < w$, then the claim is equivalent to the algebraic fact

$$s_w\Big(\sigma_w^n + \sum_{i=0}^{n} z_i w^i\Big) = s_w\Big(\sigma_w^n + 1 + \sum_{i=0}^{n} (w - 1 - z_i) w^i\Big).$$

This proof is not, however, algebraic. The claim is established by showing a 1-to-1 correspondence between the tree shapes with $\sigma_w^n + z$ nodes and those with $\sigma_w^{n+1} - z$ nodes. Select a tree with $\sigma_w^n + z$ nodes and visit each of its w^n nodes at a distance of n from the root. Each such node has w children, some are leaf nodes and the rest are λ. If a child is λ, replace it with a new leaf; if it is a leaf, replace it with λ. The modified tree has $\sigma_w^n + w^{n+1} - z = \sigma_w^{n+1} - z$ nodes. Any subtrees balanced before the transformation are balanced after and the transformation is clearly the sought-after 1-to-1 correspondence. □

4 Exact Count Formula

An exact non-recurrence formula to evaluate $s_w(z)$ is developed below. Let $n = \mathrm{L}_w(z)$, where $z > 0$, and $\mathrm{L}_w(z) = \lfloor \log_w(wz - z + 1) \rfloor - 1$. Note that $\sigma_w^n \le z < \sigma_w^{n+1}$ and $\mathrm{L}_w(z)$ is the minimum root-to-leaf path length in a balanced z node tree. Represent z uniquely as

$$z = \sigma_w^n + z', \quad \text{where } z' = \sum_{i=0}^{n} z_i w^i \text{ and } 0 \le z_i < w.$$

So $z' = z_n \ldots z_0$ is the base w representation of $z - \sigma_w^n$. Theorem 6 proves that

$$s_w(z) = \binom{w}{z_0} \prod_{i=1}^{n} \binom{w}{z_i + 1}^{\bmod(z', w^i)} \binom{w}{z_i}^{w^i - \bmod(z', w^i)}, \tag{3}$$

where $\bmod(z', w^i) = \sum_{j=0}^{i-1} z_j w^j$.

Theorem 6. *Formula* (3)*, where* $n = \mathrm{L}_w(z)$*,* $z' = z - \sigma_w^n$*, and* $z_i = \bmod(\lfloor z'/w^i \rfloor, w)$*, properly evaluates* $s_w(z)$ *when* $z > 0$.

Proof. This proof is by induction. The base step is straightforward: Let $z = \sigma_w^0 + z_0 = 1 + z_0$, where $0 \le z_0 \le w$, so $s_w(z) = \binom{w}{z_0}$ by (2). Now assume the claim is true for all $z < \sigma_w^{n+1}$ for some $n \ge 0$. Select any $\sigma_w^{n+1} \le z < \sigma_w^{n+2}$ and let $z' = z - \sigma_w^{n+1}$ so $z' = \sum_{i=0}^{n+1} z_i w^i$ where $0 \le z_i < w$. Now use (1) to expand $s_w(z)$.

$$s_w(z) = \binom{w}{z_0} s_w\Big(\sigma_w^n + \sum_{i=0}^{n} z_{i+1} w^i + 1\Big)^{z_0} \times s_w\Big(\sigma_w^n + \sum_{i=0}^{n} z_{i+1} w^i\Big)^{w - z_0} \tag{4}$$

Next, apply the inductive assumption, assuming $z_1 < w - 1$,

$$\begin{aligned} &= \binom{w}{z_0} \left[\binom{w}{z_1 + 1} \prod_{i=1}^{n} \binom{w}{z_{i+1} + 1}^{\bmod (\lfloor z'/w \rfloor, w^i) + 1} \prod_{i=1}^{n} \binom{w}{z_{i+1}}^{w^i - \bmod (\lfloor z'/w \rfloor, w^i) - 1} \right]^{z_0} \\ &\qquad \times \left[\binom{w}{z_1} \prod_{i=1}^{n} \binom{w}{z_{i+1} + 1}^{\bmod (\lfloor z'/w \rfloor, w^i)} \prod_{i=1}^{n} \binom{w}{z_{i+1}}^{w^i - \bmod (\lfloor z'/w \rfloor, w^i)} \right]^{w - z_0} \\ &= \binom{w}{z_0} \prod_{i=1}^{n+1} \binom{w}{z_i + 1}^{\bmod (z', w^i)} \binom{w}{z_i}^{w^i - \bmod (z', w^i)} \end{aligned} \tag{5}$$

which agrees with (3). When $z_1 = w - 1$ the first s_w term in (4) cannot easily be expanded by appeal to the inductive assumption though the second term can be. The first expansion, however, can easily be accomplished via Theorem 5 followed by use of the inductive assumption. Let $z'' = \sum_{i=0}^{n} (w - 1 - z_{i+1}) w^i$, then

$$\begin{aligned} s_w\Big(\sigma_w^n + \sum_{i=0}^{n} z_{i+1} w^i + 1\Big) &= s_w\Big(\sigma_w^n + \sum_{i=0}^{n} (w - 1 - z_{i+1}) w^i\Big) \\ &= \binom{w}{w - 1 - z_1} \prod_{i=1}^{n} \binom{w}{w - z_{i+1}}^{\bmod (z'', w^i)} \prod_{i=1}^{n} \binom{w}{w - 1 - z_{i+1}}^{w^i - \bmod (z'', w^i)} \\ &= \binom{w}{z_1 + 1} \prod_{i=1}^{n} \binom{w}{z_{i+1}}^{w^i - \bmod (\lfloor z'/w \rfloor, w^i) - 1} \prod_{i=1}^{n} \binom{w}{z_{i+1} + 1}^{\bmod (\lfloor z'/w \rfloor, w^i) + 1} \end{aligned}$$

Since this expansion agrees with the one at (5), the theorem follows. □

5 Discussion

Table 1 provides initial values of the sequences $s_w(z)$ for $z = 1, 2, \ldots$ where $2 \le w \le 6$ and Figures 1–6 plot $\log(s_w(z))$; logarithms are used to enhance the observable detail. Those

graphs have a somewhat fractal-like appearance and it is straightforward to see why. Let $z = \sigma_w^{n+1} + \sum_{i=0}^{n+1} z_i w^i$, $z' = z - \sigma_w^{n+1}$, low $= \mathrm{mod}(z', w^{n+1})$, and $h = z_{n+1}$. Then

$$\begin{aligned}
s_w(z) &= \binom{w}{z_0} \prod_{i=1}^{n+1} \binom{w}{z_i + 1}^{\mathrm{mod}\,(z', w^i)} \binom{w}{z_i}^{w^i - \mathrm{mod}\,(z', w^i)} \\
&= \binom{w}{z_0} \prod_{i=1}^{n} \binom{w}{z_i + 1}^{\mathrm{mod}\,(z', w^i)} \binom{w}{z_i}^{w^i - \mathrm{mod}\,(z', w^i)} \\
&\qquad \times \binom{w}{h+1}^{\mathrm{mod}\,(z', w^{n+1})} \binom{w}{h}^{w^{n+1} - \mathrm{mod}\,(z', w^{n+1})} \\
&= s_w(\sigma_w^n + \mathrm{low}) \binom{w}{h+1}^{\mathrm{low}} \binom{w}{h}^{w^{n+1} - \mathrm{low}} \\
&= \binom{w-h}{h+1}^{\mathrm{low}} \binom{w}{h}^{w^{n+1}} s_w(\sigma_w^n + \mathrm{low}).
\end{aligned}$$

Thus, the graph of $s_w(z)$, where $\sigma_w^{n+1} \le z < \sigma_w^{n+2}$ is logically broken into w pieces, each piece parameterized by h, the high order digit of z' in base w representation. For a particular h, $s_w(z) = c_1^{\mathrm{low}} c_2 s_w(\sigma_w^n + \mathrm{low})$, where c_1 and c_2 are constants and $0 \le \mathrm{low} < w^{n+1}$. Whether the shape of $s_w(\sigma_w^n + \mathrm{low})$ is stretched upwards for increasing values of low, just magnified by c_2, or stretched downwards depends on whether c_1 is greater, equal to, or less than 1, i.e., on whether $\frac{w-1}{2}$ is greater, equal to, or less than h. Each of the w pieces of this graph segment is again, a recapitulation of w magnified and possibly stretched instances of $s_w(\sigma_w^{n-1} + \mathrm{mod}(\mathrm{low}, w^{n-1}))$, and so on.

References

[1] J. A. Barnett, Sequences A110316, A131889, A131890, A131891, A131892, and A131893, 2007, at N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, http://www.research.att.com/∼njas/sequences/

[2] R. Bayer amd E. M. McCreight, Organization and maintenance of large ordered indexes, *Acta Informatica*, 1(3), pp. 173–189, 1972.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, & C. Stein: *Introduction to Algorithms,* 2^{nd} *Edition*, MIT Press and McGraw-Hill, 2001. See Chapter 13, Red-Black Trees.

[4] D. E. Knuth, *The Art of Coimputer Programming, Vol. 3, Sorting and Searching,* 2^{nd} *Edition*, Addison Wesely, 1998. See Section 6.2, Searching by comparison of keys.

[5] G. Adelson-Velskii and E. M. Landis, *An algorithm for the organization of information*, Doklady Akademii Nauk SSSR, 146 pp. 263-266, 1962 (Russian). English translation by Myron J. Ricci in Soviet Math. Doklady, 3, pp. 1259-1263, 1962.

Table 1: Some values of $s_w(z)$ for small w and z.

$s_2(\sigma_2^0), \ldots, s_2(\sigma_2^4)$
1, 2, 1, 4, 4, 4, 1, 8, 16, 32, 16, 32, 16, 8, 1, 16, 64, 256, 256, 1024, 1024, 1024, 256, 1024, 1024, 1024, 256, 256, 64, 16, 1
$s_2(\sigma_2^4), \ldots, s_2(\sigma_2^5)$
1, 32, 256, 2048, 4096, 32768, 65536, 131072, 65536, 524288, 1048576, 2097152, 1048576, 2097152, 1048576, 524288, 65536, 524288, 1048576, 2097152, 1048576, 2097152, 1048576, 524288, 65536, 131072, 65536, 32768, 4096, 2048, 256, 32, 1
$s_3(\sigma_3^0), \ldots, s_3(\sigma_3^3)$
1, 3, 3, 1, 9, 27, 27, 81, 81, 27, 27, 9, 1, 27, 243, 729, 561, 19683, 19683, 59049, 59049, 19683, 177147, 531441, 531441, 1594323, 1594323, 531441, 531441, 177147, 19683, 59049, 59049, 19683, 19683, 6561, 729, 243, 27, 1
$s_4(\sigma_4^0), \ldots, s_4(\sigma_4^2)$
1, 4, 6, 4, 1, 16, 96, 256, 256, 1536, 3456, 3456, 1296, 3456, 3456, 1536, 256, 256, 96, 16, 1
$s_5(\sigma_5^0), \ldots, s_5(\sigma_5^2)$
1, 5, 10, 10, 5, 1, 25, 250, 1250, 3125, 3125, 31250, 125000, 250000, 250000, 100000, 500000, 1000000, 1000000, 500000, 100000, 250000, 250000, 125000, 31250, 3125, 3125, 1250, 250, 25, 1
$s_6(\sigma_6^0), \ldots, s_6(\sigma_6^2)$
1, 6, 15, 20, 15, 6, 1, 36, 540, 4320, 19440, 46656, 46656, 699840, 4374000, 14580000, 27337500, 27337500, 11390625, 91125000, 303750000, 540000000, 540000000, 288000000, 64000000, 288000000, 540000000, 540000000, 303750000, 91125000, 11390625, 27337500, 27337500, 14580000, 4374000, 699840, 46656, 46656, 19440, 4320, 540, 36, 1

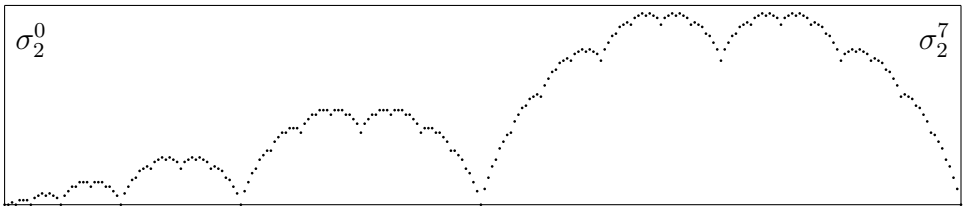

Figure 1: Initial graph of $\log(s_2(z))$

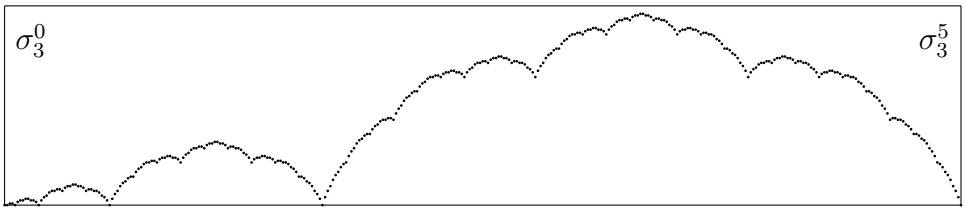

Figure 2: Initial graph of $\log(s_3(z))$

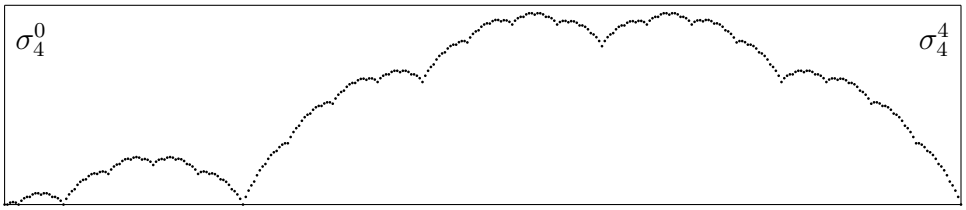

Figure 3: Initial graph of $\log(s_4(z))$

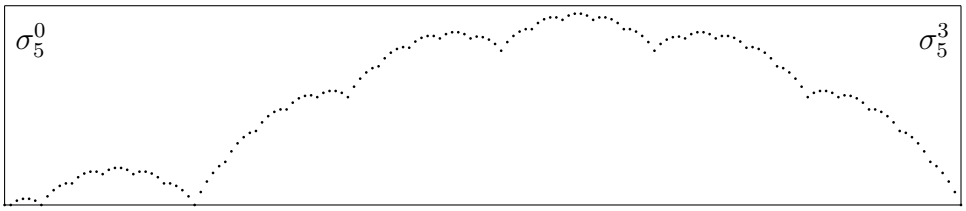

Figure 4: Initial graph of $\log(s_5(z))$

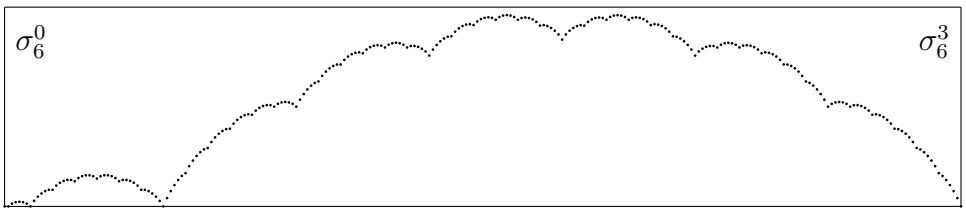

Figure 5: Initial graph of $\log(s_6(z))$

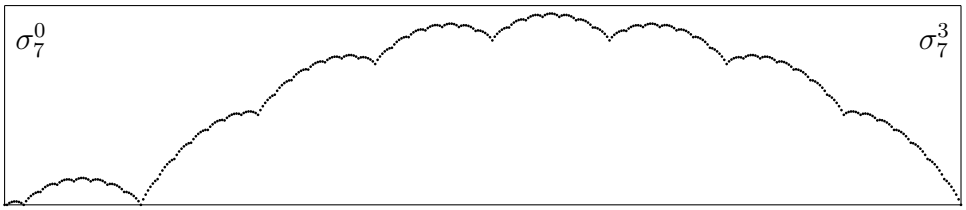

Figure 6: Initial graph of $\log(s_7(z))$