On Integer Sequences Derived from Balanced k-ary trees

SUNG-HYUK CHA
Pace University
Department of Computer Science
1 Pace Plaza, New York, NY, 10038
USA
scha@pace.edu

Abstract: This article investigates numerous integer sequences derived from two special balanced k-ary trees. Main contributions of this article are two fold. The first one is building a taxonomy of various balanced trees. The other pertains to discovering new integer sequences and generalizing existing integer sequences to balanced k-ary trees. The generalized integer sequence formulae for the sum of heights and depths of all nodes in a *complete* k-ary tree are given. The explicit integer sequence formula for the sum of heights of all nodes in a *size balanced* k-ary tree is also given.

Key–Words: complete k-ary tree, integer sequence, null-balanced k-ary tree, size-balanced k-ary tree

1 Introduction

Consider a unary ($k = 1$) tree of size n. The sum of each node's height provides an integer sequence generated by the eqn (1).

$$U(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{1}$$

This integer sequence is the famous *triagular number sequence*. The On-Line Encyclopedia of Integer Sequences [1] contains over 200,000 integer sequences. Here numerous new and generalized integer sequences from balanced k-ary tree are discovered.

A *balanced k*-ary tree is defined in several different ways [2, 3, 4, 5]. Here their relationships and taxonomy are studied. Two systematic trees whose nth tree is determined, are studied, i.e., a *complete* and *size-balanced k*-ary trees.

Adding heights or depths of every node in a *complete k*-ary tree produces an integer sequence. These are important sequences in analyzing the popular algorithms involving *d-heap* data structures. Adding heights of a *size-balanced k*-ary trees also produce new integer sequences. These sequences are very popular in numerous algorithm analysis involving the famous *divide-and conquer* paradigm.

The rest of the paper is organized as follows. Since the terminologies in *Trees*, especially the *balanced k*-ary tree, are still in flux, the section 2 provides formal definitions. In section 3 gives blah blah. Finally, the section 4 concludes this work.

2 Formal Definitions

Let n be the number of nodes in a tree, T which is the *size* of the tree, $n = |T|$. In a rooted k-ary tree, a node, t_i is either a *leaf* if it has no children or an *internal* node if it has up to k children nodes. Every node has a parent node except for one node which is called a *root*.

Definition 1 The *level* of a node is the length of the path from the node to the root.

Definition 2 The *depth* of a node is the number of the levels from the node to the root inclusively. i.e.

$$\text{depth}(t_i) = \text{level}(t_i) + 1 \tag{2}$$

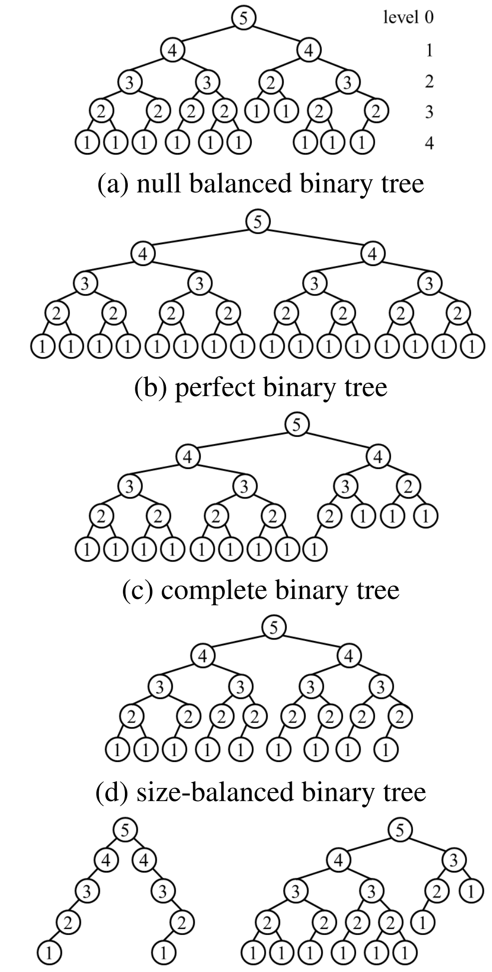
The exclusive version of the *depth* of a node is identical to the *level* as in most literatures [2, 3, 4, 5] but here it is defined inclusively. The depths of bottom level leaves in Figure 1 are 5 not 4.

Definition 3 The *height* of a node is the number of levels from the the node to the deepest leaf inclusively.

Albeit there is no universally agreed-upon definition of the height of a rooted tree [4], let h be the height of a tree which is the inclusive length of the path from the root to the deepest node in the tree. In other words, h is the height of the root node. Every node can be considered to be a root of a sub-k-ary tree and the height(t_i) is the height of the sub-k-ary tree whose root is t_i. Note that the height of a single node tree is 1 whereas it is 0 in most literatures [2, 3, 4, 5]. Here the height of an empty tree is 0; height($\varnothing$) $= 0$.

Balanced trees can be defined in various ways. Rosen defined the balanceness of a tree in terms of

ISBN: 978-1-61804-064-0



(a) null balanced binary tree

(b) perfect binary tree

(c) complete binary tree

(d) size-balanced binary tree

(e) Leaf balanced (f) Height balanced

Figure 1: balanced binary tree examples

their leaves as in Definition 4 [3].

Definition 4 A tree is called a *leaf balanced* k-ary tree if all leaves are at levels $h-1$ or $h-2$.

Binary ($k = 2$) trees in Figures 1 (a∼e) are *leaf balanced* binary trees whereas one in Figure 1 (f) is not.

A *balanced tree* is defined in terms of heights of sub-trees as in Definition 5.

Definition 5 A tree is called a *height balanced* k-ary tree if the eqn (3) is satisfied for each node t_i and for every sub-tree pair (S_x, S_y) of t_i.

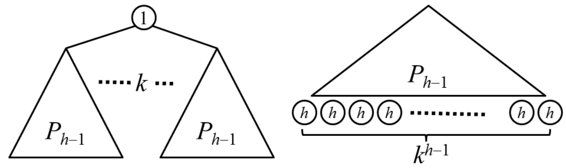
$$|\text{height}(S_x) - \text{height}(S_y)| \leq 1 \tag{3}$$

All trees in Figures 1 except for (e) are *height balanced* binary trees. A height balanced binary search tree is known as the *AVL* tree [2, 5] and the definition 5 is a generalized version of the balanced binary tree defined in [2, 5].

A different definition of a *balanced* k-ary tree is given and used in this article. It is a slight vicissitude

Table 1: size of *perfect* k-ary trees.

$k \backslash h$	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	1	3	7	15	31	63	127	255
3	1	4	13	40	121	364	1093	3280
4	1	5	21	85	341	1365	5461	21845



(a) k children recursion (b) Non-leaf level recu-

Figure 2: Two recursive relations of Perfect k-ary trees

of Definition 4. In a k-ary tree, every node has exactly k children if we consider the *null* node as a child.

Definition 6 A tree is called a *null balanced* k-ary tree if all null nodes are at levels h or $h-1$.

Fact 1 The height of the *null balanced* k-ary tree is

$$h = \lceil \log_k (n(k-1)+1) \rceil \tag{4}$$

All binary trees in Figure 1 (a∼d) are *null balanced* binary trees while both trees in Figure 1 (e,f) are not.

Definition 7 A tree is called a *perfect* k-ary tree if all internal nodes have exactly k children and all leaves lie at the same depth, h.

In case that $k = 2$ in Figure 1 (b), the perfect binary trees are possible only for $n = 1, 3, 7, 15, \cdots, 2^h - 1$.

Let P_i be size of the ith height *perfect* k-ary tree. The integer sequences of sizes of some *perfect* k-ary trees are given in Table 1. A root node has k number of *sub perfect* k-ary trees whose height is $h-1$ as shown in Figure 2 (a). Hence, P_h can be computed and defined recursively as in the eqn (5).

$$P_h = \begin{cases} 1, & \text{if } h = 0 \\ k \times P_{h-1} + 1, & \text{otherwise} \end{cases} \tag{5}$$

The tree which excludes the leaf level nodes is also a *perfect* k-ary tree as illustrated in Figure 2 (b). Hence, a non-leaf level recursive relation for $P(h)$ is given in the eqn (6).

$$P_h = \begin{cases} 1, & \text{if } h = 0 \\ P_{h-1} + k^{h-1}, & \text{otherwise} \end{cases} \tag{6}$$

The closed formula for P_h is given as follows.

$$P_h = \sum_{i=1}^{h} k^{i-1} = \frac{k^h - 1}{k - 1} \tag{7}$$

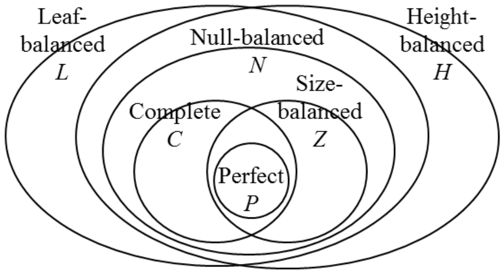


Figure 3: Venn Diagram of *balanced k*-ary trees

The *null-balanced k*-ary tree can be defined in terms of the *perfect k*-ary tree.

Definition 8 A *null-balanced k*-ary tree has a *perfect k*-ary tree whose height is $h-1$ and the remaining $n - P_{h-1}$ number of nodes are at the depth h.

There are several systematic ways to make a balanced tree for any n. Here a couple of them are considered. The first one is the *complete k*-ary tree where a node is added in the *breadth first order* as shown in Figure 1 (c).

Definition 9 A tree is called a *complete k*-ary tree if it has a *pefect k*-ary tree of height $h-1$ and the remaining nodes are added from left to right order.

In [2], the term, *complete k*-ary tree is used to refer a *perfect k*-ary tree but here it means the definition 9.

A tree can be balanced by sizes of sub-trees.

Definition 10 A tree is called a *size balanced k*-ary tree if the eqns (8) and (9) are satisfied for each node t_i and for every sub-tree pair (S_x, S_y) of t_i.

$$|\text{size}(S_x) - \text{size}(S_y)| \le 1 \tag{8}$$

$$\text{size}(S_x) \le \text{size}(S_y) \text{ if } x < y \le k \tag{9}$$

Only trees in Figures 1 (a,b,d) are *size balanced* binary trees. The sizes of k-sub trees follow the integer partition into k balanced parts defined in the eqn (10).

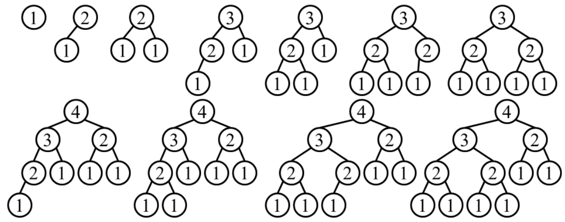
$$\text{BIP}(m,k) = \Big(\overbrace{\underbrace{\left\lceil\frac{m}{k}\right\rceil, \ldots, \left\lceil\frac{m}{k}\right\rceil}_{\tilde{k}=m\%k}, \left\lfloor\frac{m}{k}\right\rfloor, \ldots, \left\lfloor\frac{m}{k}\right\rfloor}^{k}\Big) \tag{10}$$

For examples, $\text{BIP}(23,4) = (6,6,6,5)$ and $\text{BIP}(41,5) = (9,8,8,8,8)$.

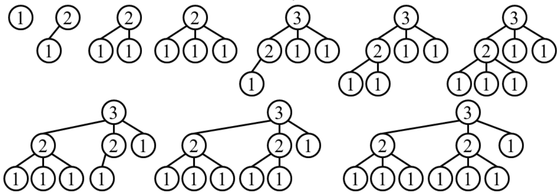
Figure 3 gives the venn diagram of *balanced k*-ary trees defined in this section.

3 Integer Sequences

Consider the first 11 and 10 sequences of complete binary and ternary trees in Figure 4 (a) and (b), respectively. Let $C(n)$ be the sum of all nodes' heights in a *complete k*-ary tree which can be computed recursively as defined in the eqn (11) as depicted in Figure 5,

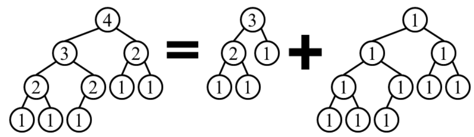


(a) binary trees

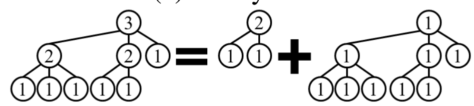


(b) ternary trees

Figure 4: *complete k*-ary tree Integer Sequences



(a) binary trees



(b) ternary trees

Figure 5: recursive relation illustration of $C(n)$

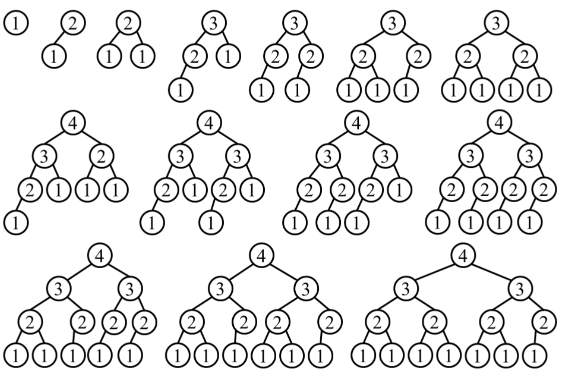
$$C(n) = \begin{cases} n, & \text{if } n \le 1 \\ C(\lceil \frac{n-1}{k} \rceil) + n, & \text{otherwise} \end{cases} \tag{11}$$

$$C'(n) = C(n) - n \tag{12}$$

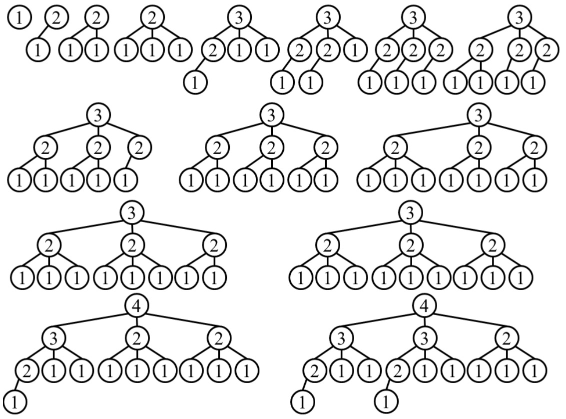
Note that the sum of exclusive heights is also defined in the eqn (12). Both $C(n)$ and $C'(n)$ integer sequences for complete binary trees are found in the *OEIS* (see [1]). However, only $C(n)$ but not $C'(n)$ is found for the complete ternary trees.

Consider the first 14 and 15 sequences of *size-balanced binary* and *ternary* trees in Figure 6 (a) and (b), respectively. Let $Z(n)$ be the sum of all nodes' heights in a em size-balanced k-ary tree. While the eqn (11) is extended from the *non-leaf level recursion* defined in the eqn (6), $Z(n)$ can be defined recursively as in the eqn (13) by slightly modifying the k *children*

ISBN: 978-1-61804-064-0



(a) binary trees



(b) ternary trees

Figure 6: *size-balanced* k-ary tree Integer Sequences

resursion defined in the eqn (5).

$$Z(n) = \begin{cases} n, & \text{if } n \leq 1 \\ h + \tilde{k} \times Z(\lceil \frac{n-1}{k} \rceil) \\ \quad +(k - \tilde{k}) \times Z(\lfloor \frac{n-1}{k} \rfloor)) , & \text{otherwise} \end{cases}$$

$$\text{where } \tilde{k} = (n-1) \mod k \tag{13}$$

Since $n-1$ may not be divisible by k, then exactly $\tilde{k} = (n-1) \mod k$ number of children's size must be one greater than the the size of other $(k - \tilde{k})$ number of children.

$$Z'(n) = Z(n) - n \tag{14}$$

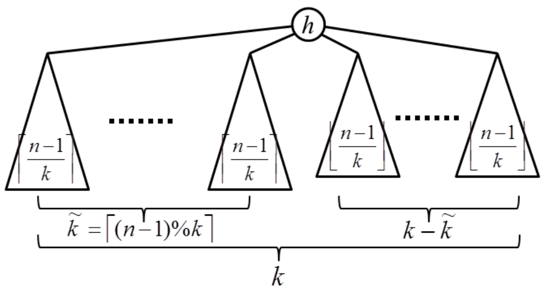


Figure 7: *size-balanced* k-ary tree Integer Sequences

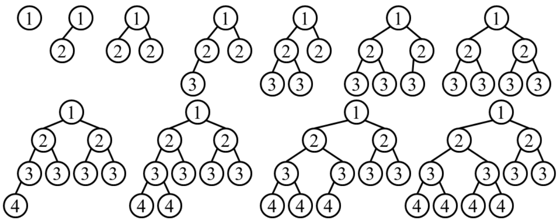
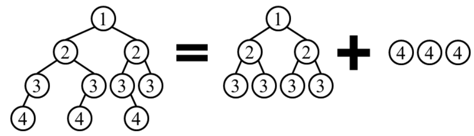
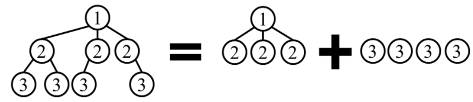


Figure 8: *null-balanced* $N_k(n)$ tree Integer Sequences



(a) binary trees



(b) ternary trees

Figure 9: Illustration of computing $N_k(n)$

Note that the sum of exclusive heights can be also defined as in the eqn (14). Neither $Z(n)$ nor $Z'(n)$ integer sequence for *size-balanced* k-ary trees appears in the *OEIS* (see [1]). However, only $C(n)$ but not $C'(n)$ is found for the complete ternary trees.

Finally, other integer sequences can be derived from aforementioned systematic k-ary trees if we add the depths instead of heights as exemplified in Figure 8. The sum of depths in a *complete* k-ary tree is the same as that in a *size-balanced* k-ary tree. In other words, any *null-balanced* k-ary tree of size n, $N(n)$ has the same sum of depths of all nodes as defined in the eqn (15).

$$N(n) = h(n - P_{h-1}) + \sum_{i=1}^{h-1} (i \times k^{i-1}) \tag{15}$$

A *null-balanced* k-ary tree has a *perfect* k-ary tree up to $h-1$ depth. The second term of the eqn (15) is adding its depth times the number of nodes in the respective depth in a *perfect* k-ary tree. And the remaining $n - P_{h-1}$ number of nodes has the value h as depicted in Figure 9.

The sum of the exclusive depth version for the *null-balanced* k-ary tree is given in the eqn (16)

$$N'(n) = N(n) - n \tag{16}$$

Both $N(n)$ and $N'(n)$ for the *null-balanced binary* tree appear in the *OEIS*. However, no integer sequences were found when $k > 2$.

ISBN: 978-1-61804-064-0

Table 2: size of *perfect* k-ary trees.

k	Name	Integer sequence for $n = 1, \ldots, 50$	$n = 1000$	OEIS
1	$U(n)$	1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136, 153, 171, 190, 210, 231, 253, 276, 300, 325, 351, 378, 406, 435, 465, 496, 528, 561, 595, 630, 666, 703, 741, 780, 820, 861, 903, 946, 990, 1035, 1081, 1128, 1176, 1225, 1275, 1275, $\cdots$	500500	A000217
2	$N(n)$	1, 3, 5, 8, 11, 14, 17, 21, 25, 29, 33, 37, 41, 45, 49, 54, 59, 64, 69, 74, 79, 84, 89, 94, 99, 104, 109, 114, 119, 124, 129, 135, 141, 147, 153, 159, 165, 171, 177, 183, 189, 195, 201, 207, 213, 219, 225, 231, 237, 243, $\cdots$	8987	A001855
	$N'(n)$	0, 1, 2, 4, 6, 8, 10, 13, 16, 19, 22, 25, 28, 31, 34, 38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78, 82, 86, 90, 94, 98, 103, 108, 113, 118, 123, 128, 133, 138, 143, 148, 153, 158, 163, 168, 173, 178, 183, 188, 193, $\cdots$	7987	A061168
	$C(n)$	1, 3, 4, 7, 8, 10, 11, 15, 16, 18, 19, 22, 23, 25, 26, 31, 32, 34, 35, 38, 39, 41, 42, 46, 47, 49, 50, 53, 54, 56, 57, 63, 64, 66, 67, 70, 71, 73, 74, 78, 79, 81, 82, 85, 86, 88, 89, 94, 95, 97, $\cdots$	1994	A005187
	$C'(n)$	0, 1, 1, 3, 3, 4, 4, 7, 7, 8, 8, 10, 10, 11, 11, 15, 15, 16, 16, 18, 18, 19, 19, 22, 22, 23, 23, 25, 25, 26, 26, 31, 31, 32, 32, 34, 34, 35, 35, 38, 38, 39, 39, 41, 41, 42, 42, 46, 46, 47, $\cdots$	994	A011371
	$Z(n)$	1, 3, 4, 7, 9, 10, 11, 15, 18, 20, 22, 23, 24, 25, 26, 31, 35, 38, 41, 43, 45, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 63, 68, 72, 76, 79, 82, 85, 88, 90, 92, 94, 96, 98, 100, 102, 104, 105, 106, 107, $\cdots$	2013	-
	$Z'(n)$	0, 1, 1, 3, 4, 4, 4, 7, 9, 10, 11, 11, 11, 11, 11, 15, 18, 20, 22, 23, 24, 25, 26, 26, 26, 26, 26, 26, 26, 26, 26, 31, 35, 38, 41, 43, 45, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 57, 57, 57, $\cdots$	1013	-
3	$N(n)$	1, 3, 5, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78, 82, 86, 90, 94, 98, 102, 106, 110, 114, 118, 122, 126, 130, 134, 138, 142, 147, 152, 157, 162, 167, 172, 177, 182, 187, 192, $\cdots$	6457	-
	$N'(n)$	0, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 106, 110, 114, 118, 122, 126, 130, 134, 138, 142, $\cdots$	5457	-
	$C(n)$	1, 3, 4, 5, 8, 9, 10, 12, 13, 14, 16, 17, 18, 22, 23, 24, 26, 27, 28, 30, 31, 32, 35, 36, 37, 39, 40, 41, 43, 44, 45, 48, 49, 50, 52, 53, 54, 56, 57, 58, 63, 64, 65, 67, 68, 69, 71, 72, 73, 76, $\cdots$	1498	A127427
	$C'(n)$	0, 1, 1, 1, 3, 3, 3, 4, 4, 4, 5, 5, 5, 8, 8, 8, 9, 9, 9, 10, 10, 10, 12, 12, 12, 13, 13, 13, 14, 14, 14, 16, 16, 16, 17, 17, 17, 18, 18, 18, 22, 22, 22, 23, 23, 23, 24, 24, 24, 26, $\cdots$	498	-
	$Z(n)$	1, 3, 4, 5, 8, 10, 12, 13, 14, 15, 16, 17, 18, 22, 25, 28, 30, 32, 34, 36, 38, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 63, 67, 71, 74, 77, 80, 83, 86, 89, 91 $\cdots$	1543	-
	$Z'(n)$	0, 1, 1, 1, 3, 4, 5, 5, 5, 5, 5, 5, 5, 8, 10, 12, 13, 14, 15, 16, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 22, 25, 28, 30, 32, 34, 36, 38, 40, 41, $\cdots$	543	-

4 Conclusion

In this paper, several different definitions of a *balanced* k-ary tree and their relationships were presented. Two kinds of special *null-balanced* k-ary trees where nth tree is determined were also presented. As shown in Table 2, explicit formulae were given to generate numerous integer sequences. Some integer sequences are already in *OEIS* but this article provided a generalized k-ary tree version formulae. The sum of height or depth integer sequences from *complete ternary* trees are not found but only the sum of inclusive height appears. These sequences are very important in the famous *d-heap* data structure.

One of the most notable findings in this paper is discovering the sum of height integer sequences from *size-balanced* k-ary trees. These sequences appear very often in certain types of the famous *divide-and conquere* algorithm analysis.

References:

[1] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. http://oeis.org/

[2] T.–H. Cormen, C.–E. Leiserson, and R.–L. Rivest, *Algorithm,* MIT Press, Cambridge, Massachusetts, 1993

[3] K.–H. Rosen, *Discrete Mathematikcs and Its Applications,* 5th ed. McGraw Hill, 2003

[4] D.–S. Malik and M.–K. Sen, *Discrete Mathematikc Structures: Theory and Applications,* Thomson Course Technology, 2004

[5] G.–M. Adel'son-Vel'skii and E.–M. Landisl, An algorithm for the organization of information., *Soviet Mathematics Doklady* 3:1259–1263, 1962.