# Integer Sequences from k-iterated Line Digraphs

Dominika Závacká<sup>1</sup>, Cristina Dalfó<sup>2</sup> and Miquel Angel Fiol<sup>3</sup>

<sup>1</sup>Comenius University, Faculty of Mathematics, Physics and Informatics, Department of Applied Informatics, Bratislava, Slovakia

<sup>2</sup>Dept. de Matemàtica, Universitat de Lleida, Igualada (Barcelona), Catalonia

<sup>3</sup>Dept. de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona Graduate School of Mathematics, and Institut de Matemàtiques de la UPC-BarcelonaTech (IMTech), Barcelona, Catalonia

#### Abstract

In this paper, we focus on integer sequences corresponding to the number of vertices in k-iterated line digraphs. We begin by introducing the core concepts related to digraphs. Then, we describe a method, proposed by Dalfó and Fiol, for calculating the order of k-iterated line digraphs. We explore various families of digraphs, such as De Bruijn, Kautz, Cyclic Kautz, and Square-free digraphs. To generate integer sequences representing the number of vertices in k-iterated line digraphs, we implement an algorithm that constructs induced subdigraphs by not allowing vertices containing forbidden subwords. The results include comparisons of the obtained integer sequences with those in the OEIS database and identification of new integer sequences. Our algorithm is implemented in the computational system *GAP*.

#### Keywords

digraph, line digraph, integer sequence, words

## 1. Introduction

This article primarily focuses on digraphs (directed graphs), which consist of vertices connected by directed edges. These directed edges indicate a one-way relationship between the vertices. By iteratively applying a specific method to obtain new digraphs, we can create a sequence of digraphs and, consequently, an integer sequence representing the numbers of vertices in these digraphs. In our work, this method involves creating line digraphs and forming sequences of *k*-iterated line digraphs.

Section 2 covers the preliminary concepts related to digraphs. We define the essential terms, such as line digraph and its iterations, regular partitions, and quotient digraphs. We also describe a method introduced by Dalfó and Fiol in [1] for computing the orders of k-iterated line digraphs. In Section 3, we present definitions and examples of some families of digraphs, including De Bruijn digraphs, Kautz digraphs, Cyclic Kautz digraphs, and Square-free digraphs. Section 4 discusses the main algorithm for obtaining integer sequences of the numbers of vertices an induced subdigraphs by removing vertices (containing forbidden subwords) from a digraph of a given family. In Section 5, we present the various

\*Corresponding author.

🛆 dominika.mihalova@fmph.uniba.sk (D. Závacká);

integer sequences that we obtained and compare them to those in the OEIS database. We list new integer sequences not found there.

## 2. Preliminaries

We introduce fundamental concepts related to digraphs, which are utilized throughout this paper. A digraph G =(V, E) consists of a (finite) set of vertices V = V(G) and a multiset of arcs (directed edges) E = E(G) between vertices of G. An arc is an ordered pair of vertices (u, v), where u is adjacent to vertex v and vertex v is adjacent from vertex u. We allow loops and multiple arcs in digraphs. A loop is an arc from vertex v to itself, that is, an arc (v, v). Multiple arcs are present in digraph G if there is more than one arc (u, v) in E(G). The in-degree of a vertex v in G, denoted  $\delta^{-}(v)$ , is the number of arcs in G adjacent to vertex v. The out-degree of a vertex vin G, denoted  $\delta^+(v)$ , is the number of arcs in G adjacent from vertex v. We say a digraph G is  $\delta$ -regular if  $\delta^{-}(v) = \delta^{+}(v) = \delta$  for all  $v \in V(G)$ . The line digraph L(G) of a digraph G is a digraph in which each vertex represents an arc of G. The vertex set of L(G) is defined as  $V(L(G)) = \{uv : (u, v) \in E(G)\}$ . Two vertices uvand wz of L(G) are adjacent if and only if v = w, meaning that the arc (u, v) in G is adjacent to arc (w, z) in G. The k-iterated line digraph  $L^k(G)$  is recursively defined as follows:  $L^0(G) = G$  and  $L^k(G) = L(L^{k-1}(G))$  for  $k \geq 0$ . A regular partition of V(G) is a partition of the vertices into m subsets  $V_1, V_2, \ldots, V_m$  such that every vertex  $v \in V_i$  is adjacent to the same number of vertices in  $V_j$ , where *i* and *j* belong to  $\{1, 2, \ldots, m\}$ . Given a digraph  ${\cal G}$  and and one of its regular partition of vertex set  $\{V_1, V_2, \ldots, V_m\}$ , a quotient matrix  $\mathcal{B}$  is an  $m \times m$ 

ITAT'24: Information technologies – Applications and Theory, September 20–24, 2024, Drienica, Slovakia

cristina.dalfo@udl.cat (C. Dalfó); miguel.angel.fiol@upc.edu (M. A. Fiol) <sup>© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License</sup>

matrix, where  $\mathcal{B}_{ij} = b_{ij}$  if there are  $b_{ij}$  arcs from partitions  $V_i$  to  $V_j$ , otherwise  $\mathcal{B}_{ij} = 0$ . A quotient digraph of digraph G, denoted  $\pi(G)$ , has its adjacency matrix equal to the quotient matrix of G.

We focus on integer sequences of the orders of the k-iterated line digraphs. Dalfó and Fiol [1] introduced a method to compute the order of k-iterated line digraph  $L^k(G)$  of digraph G. They explained that each vertex in a k-iterated line digraph is a directed walk  $v_0, v_1, \ldots, v_k$  of length k in G, where  $(v_{i-1}, v_i) \in E(G)$  for  $i = 1, \ldots, k$ . Taking the k power of the adjacency matrix  $\mathcal{A}_G$  of G, the uv-entry in  $\mathcal{A}_G^k$  corresponds to the number of k-walks from vertex u to vertex v in G. Consequently, the number of vertices  $n_k$  in  $L^k(G)$  is:

$$n_k = j \mathcal{A}_G^k j^T$$

where j = (1, ..., 1). In the case where G is a  $\delta$ -regular of order n, the  $L^k(G)$  is also a  $\delta$ -regular digraph, and the computation of its order can be simplified to:

$$n_k = \delta^k n$$

However, if G is not a  $\delta$ -regular digraph, the complexity of computing the order of  $L^k(G)$  depends completely on the dimension of  $\mathcal{A}_G$ , that is, the number of vertices in G. Dalfó and Fiol [1] introduced a method to compute  $n_k$  of  $L^k(G)$  as shown in Theorem 1. They start by obtaining a quotient matrix  $\mathcal{B}$  based on a regular partition of the vertex set of G. The size of  $\mathcal{B}$  is the same or smaller than that of  $\mathcal{A}_G$  based on the partition of vertices. The quotient matrix is used to compute the initial values  $n_k$  for the recurrence equation depending on the minimal polynomial of the quotient matrix. The subsequent values of  $n_k$  are determined by the recurrence equation.

**Theorem 1 ([1]).** Let G = (V, E) be a digraph on n vertices, and consider a regular partition  $\pi = (V_1, \ldots, V_m)$  with quotient matrix  $\mathcal{B}$ . Let  $m(x) = x^r - \alpha_{r-1}x^{r-1} - \cdots - \alpha_0$  be the minimal polynomial of  $\mathcal{B}$ . Then, the number of vertices  $n_k$  of the k-iterated line digraph  $L^k(G)$  satisfies the recurrence

$$n_k = \alpha_{r-1}n_{k-1} + \dots + \alpha_0 n_{k-r}$$
, for  $k = r, r+1, \dots$ 

initialized with the values  $n_k$ , for k = 0, 1, ..., r - 1, given by

$$n_k = \sum_{i=1}^m |V_i| \sum_{j=1}^m (\mathcal{B}^k)_{ij} = s \mathcal{B}^k j^T,$$

where  $s = (|V_1|, ..., |V_m|)$  and j = (1, ..., 1).

The recurrence equation in Theorem 1 is an efficient way of calculating the order of a k-iterated line digraph for digraphs that are not  $\delta$ -regular. Moreover, it allows us to solve other problems more effectively. The authors in

[2] apply the method by [1] to determine the number of words of length  $\ell$  over a given alphabet in some digraphs. Their approach involves constructing a digraph G that represents the connections between words of length  $\ell$  (excluding specific subwords) over the alphabet. By applying Theorem 1 to such digraphs, they determine the number of valid words. The resulting number of words of length  $\ell + k$  corresponds to the number of vertices  $n_k$  in the k-iterated line digraph of G, where G is the digraph with vertices represented by words of length  $\ell$ . We discuss the problem in the following sections.

#### 3. Some families of digraphs

Part of our research is to develop an efficient method for computing the number of words of length  $\ell$  over an alphabet of d symbols, where words do not contain any subword from a given set S. To simulate this problem on digraphs, we decided to choose families of digraphs whose vertices are represented by words over some alphabet. Each family has its specific restrictions about the words, represented by vertices and connections (arcs) between them. We swiftly introduce four known families of digraphs and show some examples.

The De Bruijn digraph  $B(d, \ell)$  has vertices labeled by all possible words  $a_1a_2 \dots a_\ell$  with  $a_i \in \{0, 1, \dots, d-1\}$ . There is an arc from vertex  $a_1a_2 \dots a_\ell$  to vertex  $a_2 \dots a_\ell a_{\ell+1}$ . An example of the De Bruijn digraph is shown in Figure 1.

The Kautz digraph  $K(d, \ell)$  has vertices labeled by all possible words  $a_1a_2 \ldots a_\ell$  with  $a_i \in \{0, 1, \ldots, d-1\}$ , where  $a_i \neq a_{i+1}$  for  $i = 1, \ldots, \ell - 1$ . There is an arc from vertex  $a_1a_2 \ldots a_\ell$  to vertex  $a_2 \ldots a_\ell a_{\ell+1}$ , whenever  $a_\ell \neq a_{\ell+1}$ . An example of a Kautz digraph is shown in Figure 2.



**Figure 1:** B(2,3) on the left and one of its quotient digraphs on the right.

The Cyclic Kautz digraph  $CK(d, \ell)$  was introduced by Böhmová, Dalfó, and Huemer in [3]. The Cyclic Kautz digraph has vertices labeled by all possible words  $a_1a_2 \ldots a_\ell$  with  $a_i \in \{0, 1, \ldots, d-1\}$ , where  $a_i \neq a_{i+1}$ for  $i = 1, \ldots, \ell - 1$ , and  $a_1 \neq a_\ell$ . There is an arc from vertex  $a_1a_2 \ldots a_\ell$  to vertex  $a_2 \ldots a_\ell a_{\ell+1}$ , whenever  $a_{\ell+1} \neq a_\ell$  and  $a_{\ell+1} \neq a_2$ . An example of a Cyclic Kautz digraph is shown in Figure 3.



**Figure 2:** K(3,3) on the left and one of its quotient digraphs on the right.



**Figure 3:** CK(3, 4) on the left with one of its quotient digraphs on the right.



**Figure 4:** SF(3,4) on the left with one of its quotient digraphs on the right.

The Square-free digraph  $SF(d, \ell)$  has vertices labeled by all possible words  $a_1a_2 \ldots a_\ell$  with  $a_i \in \{0, 1, \ldots, d-1\}$ , that does not contain an adjacent repetition of any subword of length at most 2. There is an arc from  $a_1a_2 \ldots a_\ell$  to  $a_2 \ldots a_\ell a_{\ell+1}$  when  $a_{\ell+1} \neq a_\ell$  and, if  $a_{\ell-2} = a_\ell$ , then  $a_{\ell+1} \neq a_{\ell-1}$ . An example of a Square-free digraph is shown in Figure 4.

#### 4. Algorithm

To compute the number of vertices in a k-iterated line digraph of digraph G, we decided to implement an algorithm based mostly on Theorem 1 and the method

suggested by the authors in [2]. We programmed the algorithm in the system for computational discrete algebra - GAP [4]. It is a widely used, free, and open-source system with its own programming language and various importable packages containing numerous functions. It is particularly effective for computational problems involving groups, graphs, and other combinatorial structures. For the implementation of our algorithm, we imported the packages Digraphs and GRAPE. The Digraphs package [5] was implemented to create, store, and compute various properties of digraphs. The digraph structure can be a mutable or immutable structure. The GRAPE package [6] is automatically imported with the Digraphs package. The package is intended for the construction, computation, and analysis of graphs in relation to groups. The algorithms were implemented in GAP with version 4.12.2.

The main goal of our computational method is to determine all possible integer sequences of values of  $n_k$  up to a given k, where  $n_k$  represents the number of words of length  $\ell + k$  over an alphabet of size d avoiding all possible combinations of subwords (forbidden subwords) from a set of subwords S. Initially, we employed the method described in [2] in a for-cycle and evaluated all possible combinations of forbidden subwords. However, this method was computationally very challenging as the algorithm required significant processing time to evaluate all the combinations, and it frequently produced numerous identical digraphs. To address these challenges, we opted to examine all possible induced subdigraphs instead. This alternative approach allows us to efficiently generate all integer sequences and determine the set of forbidden subwords based on forbidden and allowed vertices.

**Algorithm 1** Pseudocode: obtaining integer sequences for all subdigraphs of a given digraph G

SequencesForAllSubdigraphs(G, k <sub>max</sub> )
for all combination of $V(G)$ do
forbiddenSubwords = Difference(vertices of $G$ , com-
bination)
<pre>subdigraph = InducedSubdigraph(G, combination)</pre>
sequence = LGSequence(subdigraph, $k_{max}$ )
<pre>print (forbiddenSubwords, sequence, subdigraph)</pre>
end for
end function

Our Algorithm 1 takes two input parameters: a digraph structure (G) and  $k_{max}$ . The digraph is chosen from one of the families of digraphs discussed in Section 3, with each family imposing its own specific restrictions on the possible words and the connections between them. The vertices of digraph represent words of length  $\ell$  over an alphabet of d symbols. The parameter  $k_{max}$  specifies the maximum value of k. The algorithm begins with

a for-cycle that iterates over all combinations of vertices  $V({\cal G}),$  as this method has been demonstrated to be more effective. The set of forbidden subwords is obtained and stored in the parameter forbiddenSubwords. We construct an induced subdigraph of G based on the current combination of V(G). The subdigraph structure and the required  $k_{max}$  parameter are subsequently passed to the LGSequence() function. The function returns the integer sequence of  $n_k$  for  $k = 0, \ldots, k_{max}$ , where  $n_k$  represents the order of a k-iterated line digraph of subdigraph. In the context of the previously mentioned problem concerning the number of words of length  $\ell$  over some alphabet, the value of  $n_k$  corresponds to the number of words of length  $\ell + k$  over alphabet of d symbols avoiding subwords in forbiddenSubwords. At the end of the for-cycle, the algorithm prints a triple consisting of an example of forbidden subwords, the integer sequence with values of  $n_k$  and the subdigraph. The set of all possible combination of forbidden subwords generating the subdigraph can be computed by a separate function, which is not described here.



**Figure 5:** SF(3, 4) on the left and SF(3, 4) without subwords 021, 120 on the right.

We demonstrate our algorithm using the Square-free digraph SF(3, 4) shown in Figure 5. The input for our algorithm was the digraph SF(3, 4) with  $k_{max}$  set to 10. One of the combinations in the for-cycle included the vertices represented by the words: 0102, 0121, 0201, 1012, 1020, 1210, 2010, 2012, 2101 and 2102. We identified the forbidden subwords as 0120, 0210, 0212, 1021, 1201, 1202, 2021 and 2120 in forbiddenSubwords. These forbidden subwords can be simplified to forbidden subwords 021 and 120. The induced subdigraph is shown in Figure 5. Subsequently, we obtained the integer sequence of value  $n_k$  for  $k = 0, \ldots, 10$ , which in this case is 10, 12, 14, 18, 22, 26, 32, 40, 48, 58, 72.

#### 5. Results

We ran our algorithm on various types of digraphs discussed in Section 3. We focused on the integer sequences

#### Table 1

Forbidden subwords in the SF(3, 4) digraphs with 16 vertices and the integer sequence of the numbers of vertices  $n_k$  of kiterated line digraphs.

Forbidden subwords	Sequence
1201, 2102	16, 22, 28, 36, 46, 58, 72, 90,
2012, 2102	16, 22, 28, 38, 52, 70, 92, 124,
0120, 2120	16, 23, 31, 43, 60, 82, 112, 155,
0120, 0212	16, 23, 31, 43, 60, 83, 114, 157,
2101, 2120	16, 23, 31, 43, 61, 85, 118, 165,
1021, 1210	16, 23, 32, 45, 63, 87, 121, 170,
0102, 1201	16, 23, 32, 46, 67, 97, 139, 200,
0210, 1021	16, 23, 33, 48, 68, 96, 137, 196,
1202, 2010	16, 24, 34, 48, 68, 96, 136, 194,
0102, 0121	16, 24, 34, 48, 69, 97, 137, 196,
1020, 1202	16, 24, 34, 48, 70, 100, 142, 206,
0201, 1202	16, 24, 34, 49, 70, 100, 144, 207,
1201, 2010	16, 24, 34, 49, 71, 102, 146, 211,
0121, 1020	16, 24, 34, 50, 74, 108, 158, 232,
0102, 0212	16, 24, 35, 50, 74, 109, 158, 233,
1012, 1210	16, 24, 36, 54, 80, 120, 180, 268,
0212, 2021	16, 25, 36, 54, 81, 120, 180, 269,
0201, 1020	16, 25, 38, 59, 90, 139, 214, 329,

of the orders of k-iterated line digraphs and their presence in the database of integer sequences. Specifically, we compared the obtained integer sequences with the *OEIS* [7] database (On-Line Encyclopedia of Integer Sequences). It is a comprehensive database of integer sequences, where each sequence is uniquely identified by an ID number and accompanied by information such as definitions, references, links, and examples. We use ID numbers from OEIS database to identify the found integer sequences.

First, we applied our algorithm to some digraphs from the De Bruijn digraph family. For an alphabet of two symbols (the first non-trivial case), the number of distinct integer sequences increased as the word lengths increased. Table 2 presents all the obtained integer sequences along with examples of forbidden subwords. Additionally, we list the OEIS ID number and the type of each integer sequence.

Next, we ran the algorithm on some digraphs from the Kautz digraph family. For an alphabet of two symbols, we mostly obtained two integer sequences: A000007 and A007395 (all 2's sequence). The number of distinct integer sequences increased with an alphabet of three or more symbols.

Similarly, for digraphs from the Cyclic Kautz family, with an alphabet of two symbols, two cases occurred: no integer sequences were found if the word lengths were odd, whereas the sequences A000007 and A007395 (all 2's sequence) were found if the word lengths were even. With an alphabet of three symbols, we obtained more integer sequences, where most of them were already known.

Lastly, we ran our algorithm on some digraphs from the Square-free digraph family. Similar to the Kautz digraph family, for the alphabet of two symbols, integer sequences were found only for the digraphs SF(2, 1), SF(2, 2), and SF(2, 3). With an alphabet of three symbols, the results were more interesting as we found various integer sequences that were not in the OEIS database. For SF(3, 4), we found a total of 4947 integer sequences. Table 1 shows all integer sequences from digraphs of SF(3, 4) with 16 vertices that were not in the OEIS database.

## Acknowledgments

D. Závacká's research was partially supported by G-24-158-00 and VEGA 1/0437/23. She would like to thank her supervisor Tatiana Jajcayová for her guidance and suggestions. C. Dalfó and M. A. Fiol's research has been supported by AGAUR from the Catalan Government under project 2021SGR00434 and MICINN from the Spanish Government under project PID2020-115442RB-I00. M. A. Fiol's research was also supported by a grant from the Universitat Politècnica de Catalunya with references AGRUPS-2022 and AGRUPS-2023.

### References

- [1] C. Dalfó, M. A. Fiol, A note on the order of iterated line digraphs, Journal of Graph Theory 85 (2017) 395-399. doi:https://doi.org/10.1002/ jgt.22068.
- [2] N. H. Bong, C. Dalfó, M. A. Fiol, D. Závacká, The inner diameters of a digraph and its iterated line digraphs, 2024.
- [3] K. Böhmová, C. Dalfó, C. Huemer, The diameter of cyclic Kautz digraphs, Filomat 31 (2017) 6551–6560. doi:10.2298/FIL1720551B.
- [4] T. G. Group, GAP Groups, Algorithms, and Programming, Version 4.12.2, URL: https://www. gap-system.org, 2024.
- [5] J. De Beule, J. Jonusas, J. Mitchell, W. A. Wilson, M. Young, Digraphs, Version 1.7.1, URL: https: //gap-packages.github.io/digraphs/, 2024. Refereed GAP package.
- [6] L. H. Soicher, GRAPE, graph algorithms using permutation groups, Version 4.9.0, URL: https:// gap-packages.github.io/grape/, 2022. Refereed GAP package.
- [7] O. F. Inc., The on-line encyclopedia of integer sequences, 2024. URL: http://oeis.org.

**Table 2**Forbidden subwords in the B(2,3) digraphs, the integer sequence of the numbers of vertices  $n_k$  of k-iterated line digraphswith the number and type of sequence in the OEIS database of sequences.

Forbidden subwords	Sequence	OEIS[7]	Type of sequence
11, 10, 000	1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	A000007	$a(n) = 0^n$
00, 01, 10	1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	A000012	All 1's sequence
00, 101, 110, 111	2, 0, 0, 0, 0, 0, 0, 0, 0, 0,	A000038	$a(n) = 2 * 0^n$
11, 000, 010	2, 1, 0, 0, 0, 0, 0, 0, 0, 0,	A130713 for $n \geq 1$	a(0) = a(2) = 1, a(1) = 2; a(n) = 0 for
			n > 2
10, 000, 011	2, 1, 1, 1, 1, 1, 1, 1, 1, 1,	A054977	$a(0)=2;a(n)=1  ext{ for } n\geq 1$
00, 01	2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	A007395	All 2's sequence
00, 000, 101, 111	3, 1, 0, 0, 0, 0, 0, 0, 0, 0,	A143090 for $n \ge 5$	Aliquot sequence starting at 12
000, 001, 010, 011, 110	3, 1, 1, 1, 1, 1, 1, 1, 1, 1,	A121273 for $n \geq 4$	Number of different <i>n</i> -dimensional convex reg-
			ular polytopes that can tile $n$ -dimensional
000 001 010 011 111	2200000000		space
	3, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,	not in UEIS $114248$ for $m > 16$	The integer difference between the m
000, 001, 010, 101, 111	5, 2, 1, 0, 0, 0, 0, 0, 0, 0,	A114346 101 $n \ge 10$	dimensional unit sphere surface area minus the
			(n + 1)-dimensional unit sphere volume and
			the $(n + 2)$ -dimensional unit sphere volume
000, 001, 011, 101, 110	3, 2, 1, 1, 1, 1, 1, 1, 1, 1,	A261143 for $n \ge 1$	$a(n) = H_n(1,2)$ , where $H_n$ is the <i>n</i> -th hy-
		_	peroperator
00, 011, 101	3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	A072751 for $n \ge 14$	Greatest of the most frequent prime factors of
		_	squarefree numbers $\leq n; a(1) = 1$
01, 000	3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,	A010701	All 3's sequence
00, 010, 101	3, 4, 4, 4, 4, 4, 4, 4, 4, 4,	A113311 for $n \geq 1$	Expansion of $(1+x)^2/(1-x)$
000, 010, 011, 110	4, 2, 1, 1, 1, 1, 1, 1, 1, 1,	not in OEIS	
000, 010, 011, 111	4, 3, 1, 0, 0, 0, 0, 0, 0, 0,	A143090 for $n \geq 4$	Aliquot sequence starting at $12$
000, 011, 100, 101	4, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2,	not in OEIS	
000, 010, 011, 100	4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,	not in OEIS	
000, 001, 011, 101	4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3,	not in OEIS	
000, 001, 010, 011	4, 4, 4, 4, 4, 4, 4, 4, 4, 4,	A010709	All 4's sequence
000, 001, 011, 111	4, 5, 4, 5, 4, 5, 4, 5, 4, 5,	A010710	Periodical repetition of 4, 5
001, 010, 100, 101	4, 5, 5, 5, 5, 5, 5, 5, 5, 5,	A210032 for $n \ge 4$	a(n) = n for $n = 1, 2, 3, 4; a(n) = 5$ for
000 001 010 101	1566666666	101272  for  m > 1	$n \ge 0$ $a(n) = n$ for $n \le 6$ ; $a(n) = 6$ for $n \ge 6$
01	4, 5, 6, 7, 8, 9, 10, 11, 12	A1012/2 for $n \ge 4$	$u(n) = n \text{ for } n \leq 0, u(n) = 0 \text{ for } n > 0$ Positive integers
11,000	4 5 7 9 12 16 21 28	A000931 for $n \ge 11$	Padovan sequence
00 010	4 6 9 13 19 28 41 60	A000930 for $n \ge 5$	Naravana's cows sequence
000, 010, 011	5, 5, 5, 5, 5, 5, 5, 5, 5, 5,	A010716	All 5's sequence
000, 001, 101	5, 6, 6, 6, 6, 6, 6, 6, 6, 6,	A101101 for $n > 2$	a(1) = 1, a(2) = 5; a(n) = 6 for $n > 3$
001, 010, 011	5, 6, 7, 8, 9, 10, 11, 12,	A000027 for $n \ge 5$	Positive integers
000, 010, 111	5, 6, 7, 9, 11, 13, 16, 20,	A164317 for $n \ge 3$	Number of binary strings of length $n$ with no
		_	substrings equal to 000, 010, or 111
000, 011, 110	5, 6, 8, 10, 13, 17, 22, 29,	A052954 for $n \geq 8$	Expansion of $(2 - x - x^2 - x^3)/((1 - x) *$
			$(1 - x^2 - x^3))$
000, 010, 101	5, 7, 10, 14, 19, 26, 36, 50,	A003269 for $n \geq 8$	a(0) = 0, a(1) = a(2) = a(3) =
			1; a(n) = a(n-1) + a(n-4)
001, 010, 100	5, 7, 10, 14, 20, 29, 42, 61,	A020711	Pisot sequences $E(5,7)$ , $P(5,7)$
000, 001, 010	5, 7, 11, 16, 23, 34, 50, 73,	A164316 for $n \geq 3$	Number of binary strings of length $n$ with no
000 001 011	5 7 9 10 11 12 14 16	A001(51  for  m > 4)	Substrings equal to 000, 001, or 010
	5, 7, 8, 10, 11, 13, 14, 10,	A001031 for $n \ge 2$	Odd numbers
	5, 7, 9, 11, 15, 15, 17, 19,	$A000931 \text{ for } n \ge 12$	Padovan sequence
000,001,111	5, 8, 13, 21, 34, 55, 89, 144	A000045 for $n \ge 5$	Fibonacci numbers
000.001	6, 10, 16, 26, 42, 68, 110	A090991	Number of meaningful differential operations
	0, 10, 10, 20, 12, 00, 110, 11	11050551	of the <i>n</i> -th order on the space $R^6$
010, 011	6, 8, 10, 12, 14, 16, 18, 20,	A005843 for $n \ge 3$	Nonnegative even numbers
001, 011	6, 9, 12, 16, 20, 25, 30, 36,	A002620 for $n \ge 5$	Quarter-squares
000, 011	6, 9, 13, 18, 25, 34, 46, 62,	A164315 for $n \ge 3$	Number of binary strings of length $n$ with no
			substrings equal to 000 or 011
000, 101	6, 9, 13, 19, 28, 41, 60, 88,	A000930 for $n \geq 6$	Narayana's cows sequence
001, 010	6, 9, 14, 21, 31, 46, 68, 100,	A038718 for $n \geq 5$	Number of permutations $P$ of $n$ -set such that
			$P(1) = 1$ and $ P^{-1}(i+1) - P^{-1}(i) $ equals
		100071-	1 or 2 for $i = 1, 2,, n - 1$
001, 100	6, 9, 14, 22, 35, 56, 90, 145,	A020717	Pisot sequences $L(6, 9), E(6, 9)$
000, 010	6, 9, 15, 25, 40, 64, 104,	A006498 for $n \geq 5$	a(0) = a(1) = a(2) = 1, a(3) = 0
001	7 12 20 23 54 99 142	$\Delta 0.20732$ for $n > 1$	2, $u(n) = u(n-1) + u(n-3) + u(n-4)$ Pisot sequence $T(4, 7)$
010	7 12 21 37 65 114 200	A010901 for $n > 1$	Pisot sequences $E(4,7)$ $P(4,7)$
000	7 13 24 44 81 149 274	A282718 for $n > 4$	Tribonacci recurrence
none	8, 16, 32, 64, 128, 256, 512	A000079 for $n > 3$	Powers of 2
	., ., ., .,,,		