

The Quaternary Gray Code and Ziggu Puzzles

Madeleine Goertz 

Department of Mathematics, University of California, Berkeley, CA, USA

Aaron Williams 

Department of Computer Science, Williams College, Williamstown, MA, USA

Abstract

We investigate solutions to the new “Ziggu” family of sequential puzzles including Ziggurat, Zigguflat, Zigguhooked and so on. These puzzles have p pieces that form m mazes. We encode the state of each puzzle as a quaternary number (i.e., base 4) with $n = m + 1$ digits, where each digit gives the horizontal or vertical position in one maze. For example, the commercial version of Zigguflat has $p = 6$ pieces connected into $m = 4$ mazes and its state requires $n = 5$ digits to describe. We show that the number of states on a shortest solution is $6 \cdot 2^n - 3n - 5$ (OEIS A101946). There is only one solution of this length, and it is generated from the start configuration by a simple algorithm: make the leftmost modification that doesn’t undo the previous modification. Replacing “leftmost” with “rightmost” instead generates the unique longest solution that visits all $(3^{n+1} - 1)/2$ states (OEIS A003462). In this way, Ziggu puzzles can be viewed as 4-ary, 3-ary, or 2-ary puzzles based on how the number of state encodings, valid states, or minimum states grow with each additional maze.

Classic Gray code puzzles (e.g., Spin-Out) provide natural and illuminating comparisons. These puzzles with p pieces typically have 2^p (OEIS A000079) or $\lfloor \frac{2}{3} \cdot 2^p \rfloor$ (OEIS A000975 [21]) states on their unique (shortest) solution, and at most one modification doesn’t undo the previous modification. The states visited in a Gray code puzzle solution follow the well-known binary reflected Gray code. We show that Ziggu puzzles instead follow the quaternary reflected Gray code. More specifically, the shortest and longest solutions are both sublists of this order, meaning that some quaternary words are skipped over but the relative order of the remaining words does not change.

These results show how to solve Ziggu puzzles from the start configuration. To help solve the puzzle from an arbitrary configuration we provide $O(n)$ -time comparison, and successor algorithms, which give the relative order of two states and the next state, respectively. While Gray code puzzles have simpler recursive descriptions and successor rules, a Ziggu puzzle has a much simpler loopless algorithm to generate its shortest solution than the Gray code puzzles do. The two families are also intimately related as they have the same comparison function.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Combinatorics on words

Keywords and phrases Puzzle, Ziggu, Ziggurat, Zigguflat, Gray Code, Loopless Algorithm

Digital Object Identifier 10.4230/LIPIcs.FUN.2026.22

Related Version For additional technical results, see the preliminary version of this article.

Preliminary Version: <https://arxiv.org/abs/2411.19291> [7]

Funding The authors received financial support from NSF Grant DMS-2241623.

Madeleine Goertz: The first author would like to thank the William and Linda Frost Fund in the Cal Poly College of Science and Mathematics for their generous financial support.

Acknowledgements The authors would like to thank Williams College and Steven Miller for organizing the SMALL 2024 REU.

1 Introduction

The mathematics of puzzle solving is often categorized as recreational, but it also provides foundational educational experiences. For example, many students in Computer Science and Mathematics first learn about recursion, induction, and exponential growth through the



© Madeleine Goertz and Aaron Williams;
licensed under Creative Commons License CC-BY 4.0
13th International Conference on Fun with Algorithms (FUN 2026).
Editor: John Iacono; Article No. 22; pp. 22:1–22:22

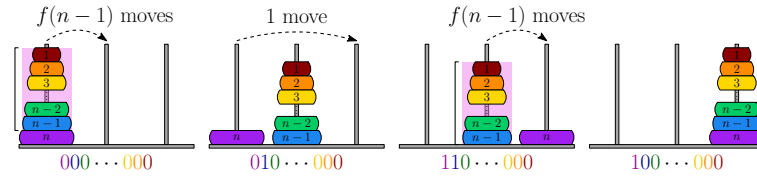


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

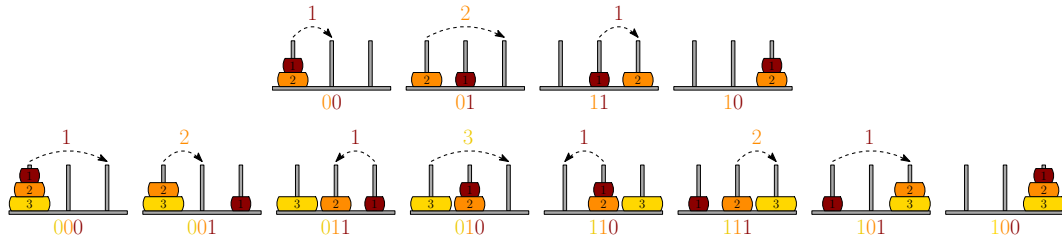
Towers of Hanoi puzzle [11]. More specifically, the standard approach of solving the puzzle involves recursively moving the smallest $n - 1$ discs, followed by moving disc n once, followed by recursively moving the smallest $n - 1$ discs again, as illustrated in Figure 1. The number of moves in this solution is given by the recurrence in (1).

$$T(n) = 2 \cdot T(n - 1) + 1 \text{ with } T(1) = 1. \quad (1)$$

Students can then use induction to prove that $2^n - 1$ is a closed formula for (1). This makes it an *exponential puzzle* since the shortest solution grows exponentially with each piece. More specifically, it is a *2-ary puzzle* as it grows like 2^n . Alternatively, it can be seen as a *3-ary puzzle* as its total number of states (and longest solution) is 3^n as seen in Figure 2a.



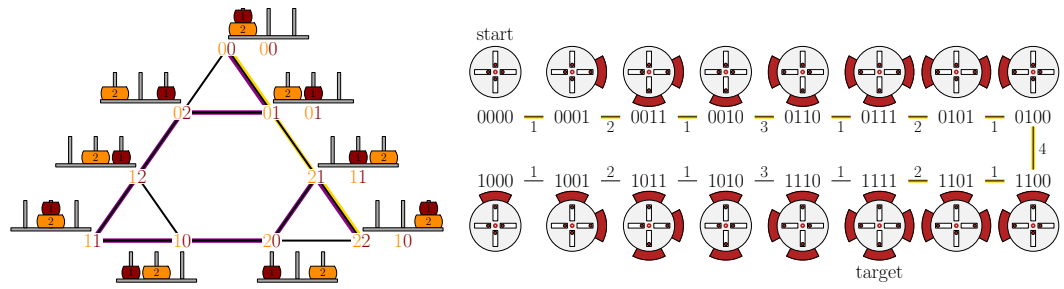
(a) Recursively move the smallest $n - 1$ discs to the middle, then move disc n to the right, then recursively move the smallest $n - 1$ discs to the right. The number of moves is $2^n - 1$ by the recurrence $f(n) = 2f(n-1) + 1$ with $f(1) = 1$, while the total number of states is 2^n by the recurrence $g(n) = 2 \cdot g(n-1)$ with $g(1) = 2$.



(b) The shortest solutions for $n = 2$ and $n = 3$ move discs 1, 2, 1 and 1, 2, 1, 3, 1, 2, 1 respectively.

■ **Figure 1** Towers of Hanoi starts with n discs on the left peg and ends with them on the right. The top disc on a peg can move to the top of another peg but not onto a smaller disc. The shortest solution is in (a) and is illustrated in (b). It is a Gray code puzzle as the binary ruler sequence gives the disc moves (where disc i moves cyclically left/right if it has the same/opposite parity as n) and the binary reflected Gray code $\text{BRGC}(n)$ gives the states if they are encoded in binary $b_n b_{n-1} \dots b_1$.

While these conclusions often mark the end of a student's investigation of the Towers of Hanoi, the puzzle can lead to much deeper discoveries in combinatorics and computation. For example, students may observe that the puzzle has 3^n states, but only 2^n of them are used in the shortest solution. There is a natural, but not immediately obvious, approach to encoding these states as an n -bit binary string. This leads to the well-known order of binary strings known as the *binary reflected Gray code* [8] and its rich generalizations [13]. Similarly, the sequence of discs that move in the shortest solution follow the *binary ruler sequence* (OEIS A001511) or more precisely the *paper folding sequence* (OEIS A164677). Given an arbitrary state of the puzzle, students may wonder if they can directly determine the next state without any additional context, or how many steps it will take to reach the solution. These questions lead to the development of the successor rules and ranking algorithms for the Gray code which are covered in *The Art of Computer Programming* [12] and elsewhere [18, 11]. In other words, the development of (1) and its closed formula are just the first lessons provided by Towers of Hanoi. Furthermore, a wide variety of other exponential puzzles have been constructed over the years that follow a similar recurrence or associated sequences. This family of puzzles is sometimes known as *Gray code puzzles*.



(a) The Towers of Hanoi state graph for $n = 2$ discs. A longest solution and the unique shortest solutions are highlighted. (b) The Brain state graph for $n = 4$ pegs. The unique solution is highlighted. There is only one move from the start state 0000 and the moves follow the ruler sequence 1, 2, 1, 3, ...

■ **Figure 2** State graphs of two Gray code puzzles (a) the Towers of Hanoi and (b) The Brain. In the Towers of Hanoi there are 3^n states and they are most naturally encoded using ternary strings (i.e., in $t_n t_{n-1} \dots t_1$ the digit t_i provides the peg on which disc i is located). A longest solution visits every state in ternary reflected Gray code order. Furthermore, discs only move between neighboring pegs so it solves the Towers of Bucharest puzzle. The unique shortest solution is a path of length 2^n . Furthermore, an alternate encoding follows the binary reflected Gray code. In the alternate encoding $b_n b_{n-1} \dots b_1$ the bit b_n is 0/1 if disc n is on the left/right peg and b_i is 0/1 if disc i is on top of disc $i + 1$ for $i < n$. The Brain is a more conventional example of a Gray code puzzle as its state graph is simply a path following the binary reflected Gray code. As a result, the only challenge from the start state 0^n to the target state 1^n is avoiding undoing your previous move and going backwards. In other words, the mechanism that enforces the solution is more interesting than the solution itself!

This paper considers the new “Ziggu” family of exponential puzzles and various facets involved in solving them. For example, this family includes the Ziggustretch puzzle which is illustrated in Figure 3. As we will see, the shortest solution to all of the puzzles in the family follow a similar but more complicated recurrence.

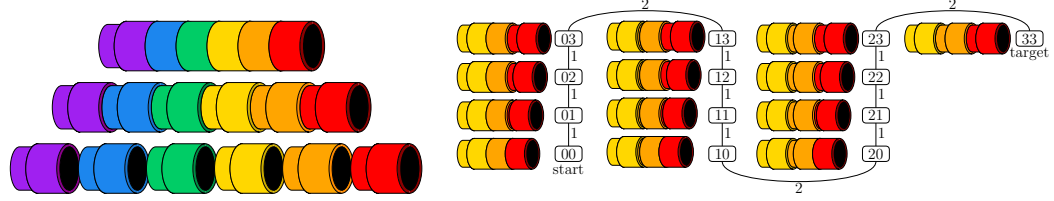
$$Z(n) = 3 \cdot Z(n-1) - 2 \cdot Z(n-2) + 3 \text{ with } Z(1) = 4 \text{ and } Z(2) = 13. \quad (2)$$

Roughly speaking, solving a puzzle with $m = n - 1$ Ziggu mazes requires solving its first $m - 1$ Ziggu mazes three times, however, the solution to its first $m - 2$ Ziggu mazes can be skipped in the latter two sub-solutions; see Figure 12b. The recurrence in (2) has a closed form of $Z(n) = 6 \cdot 2^n - 3n - 5$ (OEIS A101946), whose dominant 2^n term makes it a *2-ary exponential puzzle* in terms of its shortest solution. Just like the Towers of Hanoi, we’ll see that the puzzles can also be viewed as 3-ary puzzles as they have $(3^{n+1} - 1)/2$ (OEIS A003462) total states and solutions of this length. In fact, we’ll also view them as 4-ary puzzles due to an intimate connection with the quaternary (base-4) reflected Gray code.

1.1 New Results

Those who have held a Ziggu puzzle may have stumbled upon this approach to exploring its configurations: modify the leftmost piece that does not undo the most recent modification. Pleasantly, this approach will eventually solve the puzzle. Moreover, it does so in as few moves as possible. However, it isn’t immediately clear how many moves are made during the solution. We prove that exactly $6 \cdot 2^n - 3n - 5$ states are on the shortest solution, and we refer to these states as *Ziggu states* or strings. Furthermore, if one replaces “leftmost” with “rightmost” then the exploratory approach again solves the puzzle, and it does so in as many moves as possible. In fact, this longest solution visits every one of the $(3^{n+1} - 1)/2$ states, and it is the only solution to do so. Both of these formulae assume that state is

encoded as one of the 4^n quaternary (i.e., base 4) strings with n digits, as explained later on. These three formulae are summarized in Table 4 and they explain why Ziggu puzzles can be considered to be 4-ary, 3-ary, or 2-ary exponential puzzles.



(a) The puzzle starts with the pieces screwed together (state 0^n) (top) and ends with them as far apart as possible (state 3^n) (middle) which unlocks them (bottom). (b) State space for $p = 3$ pieces. States are $n = 2$ quaternary digits q_2q_1 which encode the distance between adjacent pieces. Edges indicate the valid moves and are labeled by the index of the digit changed by ± 1 . The graph is $m = 1$ Ziggu maze.

■ **Figure 3** The goal of Ziggustretch is to unscrew p interlocking pieces and take them apart. Adjacent pieces are separated by a discrete distance between 0–3, so the state can be encoded as a quaternary (base-4) word with $n = p - 1$ digits. Hidden channels in the cylinders ensure that only one gap can be changed at a time. Furthermore, the moves are physically restricted based on $m = p - 2$ “ziggu” mazes. For example, when there are $p = 3$ pieces the state is $n = 2$ quaternary digits and the state space is a path that follows $m = 1$ maze. The puzzle is more interesting with additional pieces as multiple mazes must be navigated and “shortcuts” are possible (see Figure 4).

These results imply that Gray code puzzles are simpler than Ziggu puzzles. This is because Gray code puzzles can typically be solved using a similar exploratory approach: modify the only piece that does not undo the most recent modification. In fact, in most Gray code puzzles there is never a choice of which piece to modify (so long as the previous modification is known) and this approach gives the only solution.

We show that the connection between Ziggu puzzles and Gray code puzzles runs even deeper. In Gray code puzzles each state can be represented by an n -bit binary string, and the next state is obtained by flipping the bit according to the binary reflected Gray code. Similarly, the longest and shortest solutions to Ziggu puzzles can be understood using the quaternary (i.e., base 4) reflected Gray code. More specifically, the longest solution is obtained by skipping over invalid encodings in the quaternary reflected Gray code, and the shortest solution is obtained by additionally skipping over non-Ziggu states in the longest solution. This is illustrated in Figure 4 and the gaps in Table 1.

The aforementioned results show how to solve Ziggu puzzles from the start configuration to the target configuration in one sitting. However, a solver may wish to leave the puzzle in a state that is part way through the longest or shortest solution. To help with these situations we provide $O(n)$ -time comparison and successor algorithms, which give the relative order of two states and the next state, respectively.

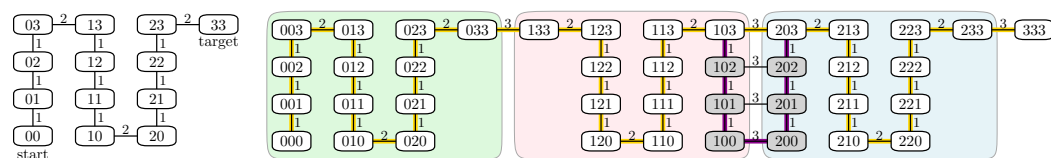
A preliminary version of this paper has additional material [7] including ranking and unranking (see Section 11). Its longer title also pays homage to an article by Martin Gardner [5] (see also [6]) which discusses the Gray code and several of the aforementioned puzzles.

1.2 Outline

Section 2 provides background on Gray code and Ziggu puzzles. Section 3 gives a non-technical description of how to solve Gray code and Ziggu puzzles. Section 4 explains our Ziggu maze abstraction and how the mazes are formed in Ziggufat and Zigguhooked. Section 5 derives

■ **Table 1** For $n = 3$ digits, we examine the quaternary reflected Gray code Q_n , the longest solution V_n , and the shortest solution S_n . For each order, we include the ranking, the Gray code, the change sequence, and the unsigned change sequence. The last two columns present the maze representation corresponding to each 2-digit base 4 string and the dials representation for the Zigguhooked puzzle. More information on ranking can be found in the preliminary version of this paper [7].

Quaternary Gray code				Longest solution				Shortest solution				Mazes	
rank	Q_3	ΔQ_3	$\Delta^\pm Q_3$	rank	V_3	ΔV_3	$\Delta^\pm V_3$	rank	S_3	ΔS_3	$\Delta^\pm S_3$	maze	dials
0	000	1	1	0	000	1	1	0	000	1	1	000	000
1	001	1	1	1	001	1	1	1	001	1	1	001	001
2	002	1	1	2	002	1	1	2	002	1	1	002	002
3	003	2	2	3	003	2	2	3	003	2	2	003	003
4	013	1	-1	4	013	1	-1	4	013	1	-1	013	013
5	012	1	-1	5	012	1	-1	5	012	1	-1	012	012
6	011	1	-1	6	011	1	-1	6	011	1	-1	011	011
7	010	2	2	7	010	2	2	7	010	2	2	010	010
8	020	1	1	8	020	1	1	8	020	1	1	020	020
9	021	1	1	9	021	1	1	9	021	1	1	021	021
10	022	1	1	10	022	1	1	10	022	1	1	022	022
11	023	2	2	11	023	2	2	11	023	2	2	023	023
12	033	1	-1	12	033	3	3	12	033	3	3	033	033
13	032	1	-1										
14	031	1	-1										
15	030	3	3										
16	130	1	1										
17	131	1	1										
18	132	1	1										
19	133	2	-2	13	133	2	-2	13	133	2	-2	133	133
20	123	1	-1	14	123	1	-1	14	123	1	-1	123	123
21	122	1	-1	15	122	1	-1	15	122	1	-1	122	122
22	121	1	-1	16	121	1	-1	16	121	1	-1	121	121
23	120	2	-2	17	120	2	-2	17	120	2	-2	120	120
24	110	1	1	18	110	1	1	18	110	1	1	110	110
25	111	1	1	19	111	1	1	19	111	1	1	111	111
26	112	1	1	20	112	1	1	20	112	1	1	112	112
27	113	2	-2	21	113	2	-2	21	113	2	-2	113	113
28	103	1	-1	22	103	1	-1	22	103	3	3	103	103
29	102	1	-1	23	102	1	-1						
30	101	1	-1	24	101	1	-1						
31	100	3	3	25	100	3	3						
32	200	1	1	26	200	1	1						
33	201	1	1	27	201	1	1						
34	202	1	1	28	202	1	1						
35	203	2	2	29	203	2	2	23	203	2	2	203	203
36	213	1	-1	30	213	1	-1	24	213	1	-1	213	213
37	212	1	-1	31	212	1	-1	25	212	1	-1	212	212
38	211	1	-1	32	211	1	-1	26	211	1	-1	211	211
39	210	2	2	33	210	2	2	27	210	2	2	210	210
40	220	1	1	34	220	1	1	28	220	1	1	220	220
41	221	1	1	35	221	1	1	29	221	1	1	221	221
42	222	1	1	36	222	1	1	30	222	1	1	222	222
43	223	2	2	37	223	2	2	31	223	2	2	223	223
44	233	1	-1	38	233	3	3	32	233	3	3	233	233
45	232	1	-1										
46	231	1	-1										
47	230	3	3										
48	330	1	1										
49	331	1	1										
50	332	1	1										
51	333	2	-2	39	333			33	333			333	333
52	323	1	-1										
53	322	1	-1										
54	321	1	-1										
55	320	2	-2										
56	310	1	1										
57	311	1	1										
58	312	1	1										
59	313	2	-2										
60	303	1	-1										
61	302	1	-1										
62	301	1	-1										
63	300												



(a) State graph for $n = 2$. (b) State graph for Ziggu puzzles encoded with $n = 3$ digits.

■ **Figure 4** Ziggu puzzles with n digits have the same state space. In (a), the shortest and longest solutions are the same and consist of 13 states. In (b) the gold edges form the unique shortest solution while the additional purple edges form the unique longest solution. The gray vertices appear on the longest solution but not the shortest solution. The lengths of the shortest and longest paths are (a) 13 and 13 and (b) 34 and 40, respectively, as in Table 4. Note that (a) is a subgraph of (b) (with 0 prefixed to each state label).

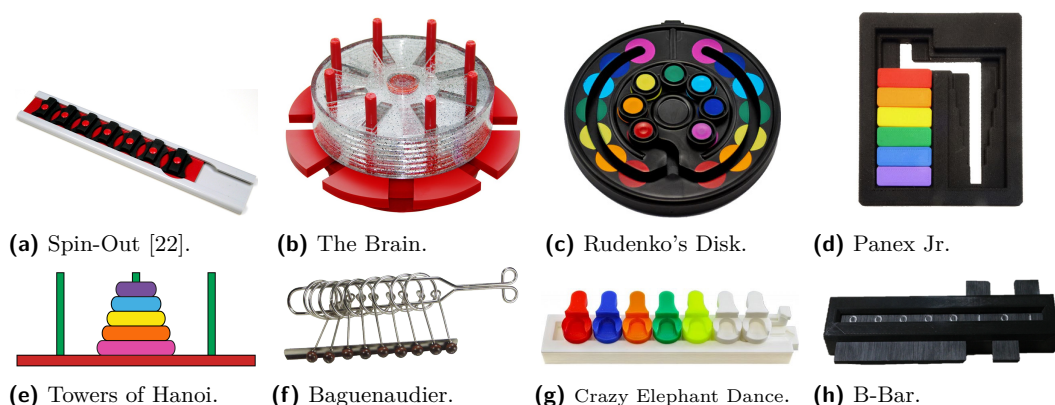
recursive formulae for shortest and longest solution lengths. Section 6 cover the binary and quaternary reflected Gray codes. Section 7 provides recursive descriptions of Gray code and Ziggu puzzle solutions. Sections 8 and 9 present the successor rules and comparison formulae for the solutions to these puzzles. Section 10 presents loopless algorithms for generating the shortest solution to Ziggu puzzles. We end with final remarks in Section 11. Omitted proofs follow from first principles or appear in [7].

2 Sequential Puzzles

This section discusses two families of sequential puzzles: Gray code puzzles and Ziggu puzzles. For a broader catalog of n -ary puzzles see Goetz Schwandtner's *Extremely Puzzling!* site [19].

2.1 Gray Code Puzzles: Chinese Rings, Towers of Hanoi, Spin-Out ...

The most famous family of exponential puzzles is the *Gray code puzzle* family, which are related to the well-known pattern of binary numbers or words known as the *binary reflected Gray code* (BRGC) or simply “the” *Gray code* [13]. A sampling appears in Figure 5.



■ **Figure 5** The Gray code family of puzzles.

The family contains vintage puzzles including Spin-Out (see [16] and [3]) and The Brain. In these puzzles the underlying state space or state graph is simply a path labeled by the Gray code. The ends of the path are 0^n and 10^{n-1} and correspond to the first and last words

■ **Table 2** The number of pieces n in various Gray code puzzles, along with the number of states in the unique (shortest) solutions. Towers of Hanoi is ternary (3-ary) in terms of the total number of states, but only binary (2-ary) in terms of its shortest solution.

puzzle	base	pieces n	start state	target state	shortest solution
Spin-Out	binary	7	1111111	0000000	86
The Brain	binary	8	00000000	11111111	171
B-Bar	binary	8	00000000	10000000	256
Baguenaudier	binary	varies	1^n	0^n	OEIS A005578
Towers of Hanoi	binary ternary	varies	0^n	10^{n-1}	2^n

in the Gray code. The state 1^n also features prominently as it is the start or target state in many of the puzzles. For example, the state space of The Brain with $n = 4$ pieces appears in Figure 2b and the state space for Spin-Out with $n = 3$ dials appears in Figure 11a.

Most famously, the family contains two historical puzzles. In the Towers of Hanoi the Gray code provides the shortest solution of length 2^n among its 3^n states, and in Baguenaudier (or “Chinese Rings”) [17] the state space is the Gray code except that some pairs of moves can be made simultaneously.

Perhaps the most direct implementation of a Gray code puzzle is B-Bar by Felix Davis. As in The Brain, its state space is the path formed by $\text{BRGC}(n)$, however, in this case the puzzle begins and ends at the two extreme nodes. More broadly, Hamilton paths have been used as the basis for other sequential puzzles, including Hamilton’s Icosian game [15, 14].

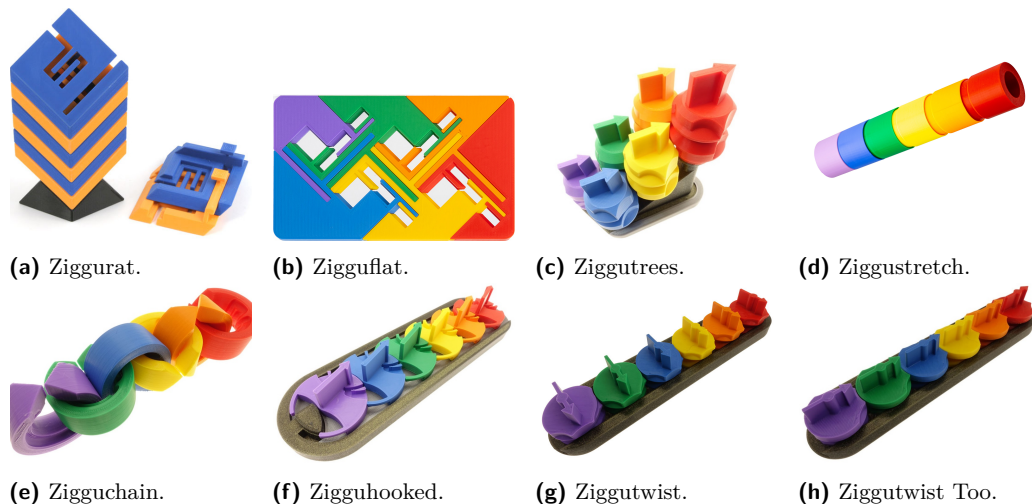
The Gray code puzzle family also includes puzzles in which solutions are variations of the Gray code. For example, Rudenko’s Disc can be seen as a variation of the Towers of Hanoi. More specifically, the rule that larger discs cannot be placed on top of smaller discs is replaced by a physical limitation that disc i can only be pushed down i positions from the top of each channel. In other words, disc i can have at most $i - 1$ discs on top of it, but the discs on top can have any number. As a result, the binary reflected Gray code provides a solution, but not the shortest. Panex Jr and Stacker by CoreMods further refine this idea by limiting the height of one of the channels (i.e., pegs) thereby lengthening the solutions.

The Towers of Bucharest restricts the Towers of Hanoi by not allowing discs to move directly between the left and right pegs. The puzzle is also known as Linear Towers of Hanoi. Regardless of its name, the unique solution visits all 3^n states in ternary Gray code order [20, 10, 11], and so it is also the longest possible solution to the Towers of Hanoi (see Figure 2). Another puzzle that follows the ternary Gray code is Crazy Elephant Dance [2].

2.2 Ziggu Puzzles: Ziggurat, Zigguflat, Zigguhooked ...

The Ziggurat puzzle was created by Eitan Cher and Bram Cohen and submitted to the 2021 Puzzle Design Competition [1]. The puzzle is notable for several reasons. It is an *exponential puzzle* since it consists of some number of (nearly) identical pieces and the number of steps required to solve the puzzle grows exponentially in some base. Unlike previous puzzles of this type it is a *burr* meaning that the goal is to fully disassemble (or reassemble) it. It is also *frameless* meaning that the puzzle consists only of its pieces without any other structure

holding it together.¹ The name of the puzzle is based on how the pieces are initially stacked into a 3-dimensional pyramidal configuration resembling a ziggurat temple. The underlying maze mechanic underpinning the puzzle is discussed in Section 4.



■ **Figure 6** The Zigu family of puzzles. Each puzzle has $p = 6$ pieces in alternating or rainbow colors. Some puzzles also have a black stand or frame which does not count as a piece.

Oskar van Deventer implemented the same underlying maze structure in a variety of creative ways in subsequent puzzles including Ziguflat, Ziggutwist, Ziggutwist Too, Zigguchain, Ziguhooked, Ziggutrees, and Ziggustretch. All of these puzzles are seen in Figure 6 with $p = 6$ pieces. In some of these puzzles the mazes can be clearly seen (e.g., Ziguflat) while in others they are partially obscured (e.g., Ziggutwist) or fully obscured (e.g., Ziggustretch). Zigu puzzles are typically color-coded so that the red piece is the only one that can be modified in the starting configuration. Right-handed solvers often hold the puzzle with this piece on the right. We'll assume this orientation in our discussions, and we'll also refer to the red piece as the first piece, the orange piece as the second piece, and so on. (In Ziggurat the bottom piece is the first piece.)

While each Zigu puzzle is unique, they fall into two primary categories, summarized in Table 3. Ziggurat, Ziguflat, Ziggutrees, and Ziggustretch (Figures 6a-6d) are all *4-neighbor* puzzles, meaning that each piece touches at most four other pieces. In particular, Figure 6b shows that the yellow piece touches the orange and red pieces in one direction, as well as the blue and green pieces in the other direction. On the other hand, Zigguchain, Ziguhooked, Ziggutwist, and Ziggutwist Too are all *2-neighbor* puzzles, as each piece touches up to two other pieces. This is easily seen in Figures 6e-6h, where the pieces are arranged linearly and each piece touches its left and right neighbor.

In the remainder of this paper, we consider Ziguflat and Ziguhooked as examples of 4- and 2-neighbor puzzles. These have the advantage of using distinctly different implementations of the Zigu maze while still being planar, which makes the mechanics visible. As noted in Table 3, while both Ziguflat and Ziguhooked consist of $p = 6$ pieces, Ziguhooked will take longer to solve since it has 5 mazes, while Ziguflat has only 4 mazes.

¹ Note that the black base in Figure 6a is simply a stand and is not part of the puzzle.

■ **Table 3** There are two types of commercially available Ziggu puzzles. Each piece either interacts with the previous and next pieces (2-neighbor) or the previous two and next two pieces (4-neighbor). This affects how many mazes that must be solved, the number of quaternary digits we use to represent an individual state, and the number of states in their shortest and longest solutions.

puzzles	type	pieces p	mazes m	digits n	shortest solution	longest solution
Ziggurat, Ziggufat, Ziggutrees, Ziggustretch	4-neighbor	6	4	5	172	364
Zigguchain, Zigguhooked, Ziggutwist, Ziggutwist Too	2-neighbor	6	5	6	361	1093

■ **Table 4** Integer sequences related to Ziggu puzzles encoded by n digits. Zigguhooked with $p = 6$ dials is encoded by $n = 6$ digits, so there are $4^6 = 4096$ state encodings. Only 1093 of the encodings represent valid states, and 361 are on a shortest solution. In particular, $132222 \in \mathbb{Q}_6$ is an encoding, but it is not a valid state as the dials cannot be physically rotated as $\rightarrow \nearrow \rightarrow \rightarrow \rightarrow \rightarrow$. So this encoding only contributes to the top row's quantity as $132222 \notin \mathbb{V}_6$ and $132222 \notin \mathbb{S}_6$.

strings	set	cardinality	growth	values	sequence
encoding	$\mathbb{Q}_n$	4^n	$\theta(4^n)$	4, 16, 64, 256, 1024, 4096 , ...	OEIS A000302
valid state	$\mathbb{V}_n$	$(3^{n+1} - 1)/2$	$\theta(3^n)$	4, 13, 40, 121, 364, 1093 , ...	OEIS A003462
ziggu	$\mathbb{S}_n$	$6 \cdot 2^n - 3n - 5$	$\theta(2^n)$	4, 13, 34, 79, 172, 361 , ...	OEIS A101946

Despite these seemingly large differences, all of the Ziggu puzzles are identical in an abstract sense. More specifically, all Ziggu puzzles with m mazes have identical state spaces and solutions.

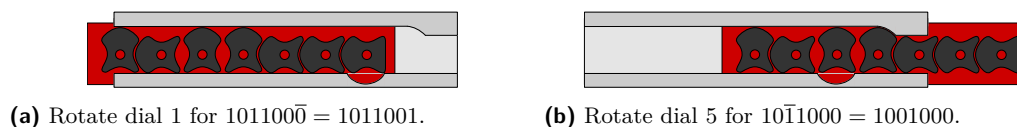
3 Solving Puzzles from the Start

In this section, we give simple approaches to solving Gray code puzzles and Ziggu puzzles from their start states. The content is tailored to those who wish to solve a puzzle once as simply as possible. We assume the solver begins from the start state and knows their previous move. Subsequent sections in this paper formalize these results and address more nuanced questions and scenarios.

3.1 Don't Look Back!

Gray code puzzles and Ziggu puzzles can be solved using the same basic principle: don't look back! This means that you should never reverse your progress by undoing your most recent move. This advice is enough to solve many Gray code puzzles due to the following: *Gray code puzzles have at most one non-reversing move*. In other words, when you move to a new state there is at most one move that doesn't take you back to your previous state. In some puzzles (e.g., Spin-Out) the only complication is that there are two different moves that can be made from the start state; the wrong first move leads toward a dead-end from which you need to undo your final move and *then* never look back. That is, the state space is a path and only one direction takes you to the solution. For example, the neighbors in

Spin-Out² are illustrated in Figure 7. In Towers of Hanoi there can be two non-reversing moves, but this reduces to one when putting additional restrictions on how and when to move the smallest disc.³



■ **Figure 7** Spin-Out states are encoded as a binary string $b_n b_{n-1} \cdots b_1$ based on the orientation of its n dials. Horizontal translation is ignored, so (a) and (b) are both encoded as $b_7 b_6 \cdots b_1 = 1011000$. The contours of the top and bottom rails ensure that at most two dials can be rotated in any given state. These include the rightmost dial b_1 as in (a), and the dial to the left of the rightmost vertical dial as in (b). In particular, b_1 or b_2 can be changed from its start state $b_7 b_6 \cdots b_1 = 1111111$, with the former change leading in the wrong direction (cf., Figure 2b).

Ziggu puzzles are more challenging than Gray code puzzles due to the following: *Ziggu puzzles can have more than one non-reversing move*; hence *Ziggu puzzles have solutions of different lengths*. A puzzle with m mazes and $n = m + 1$ digits can have up to $\lceil \frac{n}{2} \rceil + 1$ non-reversing moves^{4,5}. Figure 8 gives an example of such a state in Zigguhooked. While Ziggu puzzles have more complicated states, the unique shortest and longest solutions are generated by simple extremal versions of the basic don't look back principle: For the **shortest/longest** solution, always make the **leftmost/rightmost** non-reversing move. Since there is only one available move from the start state, you always begin by moving in the correct direction. Besides the start state, the only other “dead end” is the solution state. As a result, you'll never need to backtrack over previous moves. However, you can return to a previous state by going around in a circle (i.e., through a cycle of states) if you don't follow either of the extremal versions of the basic principle.

3.2 One Bit to Remember: Resuming a Shortest Solution

By design exponential puzzles take a long time to solve, and solvers may wish to take a break. Trouble will ensue if a solver resumes by undoing the last move they made before taking a break. This can be avoided by recording the last move made. In fact, less information is needed: we only one bit of information. One literal bit.

Gray code puzzles have a “metronome” property wherein every second change involves the first piece. The change sequence follows the binary ruler sequence: 1, 2, 1, 3, 1, 2, 1, 4, 1, ... (OEIS A001511). So you only need to remember if your most recent move involved the first piece or not. The metronome property holds for puzzles that start in an intermediate state (e.g., Spin-Out) although the change sequence is a (reversed) subsequence of OEIS A001511.

The shortest solution to Ziggu puzzles instead have a “stair-climbing” property wherein successive changes differ by at most one: 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 3, 2, 1, ... Suppose that you always make the 1 moves in succession (i.e., when changing the first piece you do it three

² A design flaw in the initial run of Spin-Out puzzles introduces more states and shorter solutions by allowing dials on the right to be rotated to the right. This can be fixed by extending the right end of the frame. Our discussion of Spin-Out ignores the flaw.

³ On every odd numbered turn, move disc 1 one peg to the left (right) for even (odd) n .

⁴ For example, the state 2020...202 (see Section 4) allows changes to each 2 and the last 0.

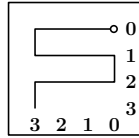
⁵ The shortest solution only visits states with up to two non-reversing moves. In this sense, Ziggu puzzles remain relatively simple until you go off course.



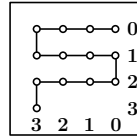
(a) In **10203** the digit **0** is locked by the **2** digit to its right, but the other four digits can change. In the corresponding Ziguhooked state the **green** dial cannot rotate in either direction due to the orientation of the **yellow** dial, but all of the other dials can rotate in one direction.



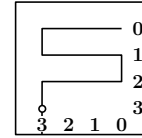
■ **Figure 8** In Zigu puzzles there are states in which up to $\lceil \frac{n}{2} \rceil + 1$ digits can be changed. This is illustrated in (a) by a Ziguhooked state in which $\lceil \frac{5}{2} \rceil + 1 = 4$ of the $n = 5$ dials can be changed. Digits that can change can only be changed in one way (i.e., incremented or decremented). These changes are shown in (b)–(e) with overlines and underlines indicating increments (counterclockwise rotation) and decrements (clockwise rotation), respectively. Note that (a) would not be encountered on the shortest solution since states on the shortest solution have at most three digits that can be changed (regardless of n). (An exception is the rightmost **red** digit, which can be freely incremented or decremented when it is not **0** or **3**.)



(a) Initial state.



(b) 13 valid locations.



(c) Maze exit.

■ **Figure 9** (a) The standard Zigu maze is on a 4-by-4 grid and its position is initially location (0, 0). (b) There are 13 valid locations: $(3, i)$ is invalid for $0 \leq i \leq 2$. (c) The maze's exit is accessible from location (3, 3) (but only if higher mazes have been exited).

times in a row). Treating each 1, 1, 1 as a single 1 is natural in most of the Zigu puzzles (e.g., the first dial in Ziguhooked moves freely between its two extreme states 0 and 3). With this assumption you just remember if your most recent move went up (e.g., 2 then 3) or down (e.g., 3 then 2). In the first case you will attempt to move up again (e.g., 2 then 3 then 4); if the puzzle does not allow this, then you will move down (e.g., 2 then 3 then 2). In the second case you move down again (e.g., 3 then 2 then 1) or repeatedly move the first piece.

Here is a general tip to avoid undoing your work in a Gray code or Zigu puzzle: Never pause after changing the first piece or going down.

4 Zigu Mazes, Pieces, and Digits

While the Zigu family consists of a wide family of seemingly distinct puzzles, they all use the same underlying mechanic of the Zigu maze.

4.1 Zigu Mazes

► **Definition 1.** A Zigu maze is the S-shaped maze in Figure 9. It lives on a 4×4 grid, and the 13 valid positions are

$$\{(i, j) \mid i \in \{0, 1, 2\}, j \in \{0, 1, 2, 3\}\} \cup \{(3, 3)\}. \quad (3)$$

The rows and columns are labeled 0, 1, 2, 3 from top to bottom and from right to left, respectively. The 13 valid states are connected as shown in Figure 9. The initial state of the maze is (0, 0), and the maze's exit is accessible from (3, 3).

An m -Ziggu maze is a sequence of m Ziggu mazes in states

$$(r_1, c_1), (r_2, c_2), \dots, (r_m, c_m) \text{ where } r_i = c_{i+1} \text{ for all } i = 1, 2, \dots, m-1. \quad (4)$$

That is, the horizontal position of a maze is the vertical position of the maze to its right.

The base 4 representation of an m -Ziggu maze is the length $n = m + 1$ base 4 string $c_1 r_1 r_2 \cdots r_m$.

For example, the 3-Ziggu maze with state (0, 1), (2, 0), (3, 2) has the maze representation $\Xi\Xi\Xi$. While there are 6 digits in the state description, since $r_1 = c_2$ and $r_2 = c_3$, we can describe the state of this 3-Ziggu maze with 4 digits with the base 4 string 1023.

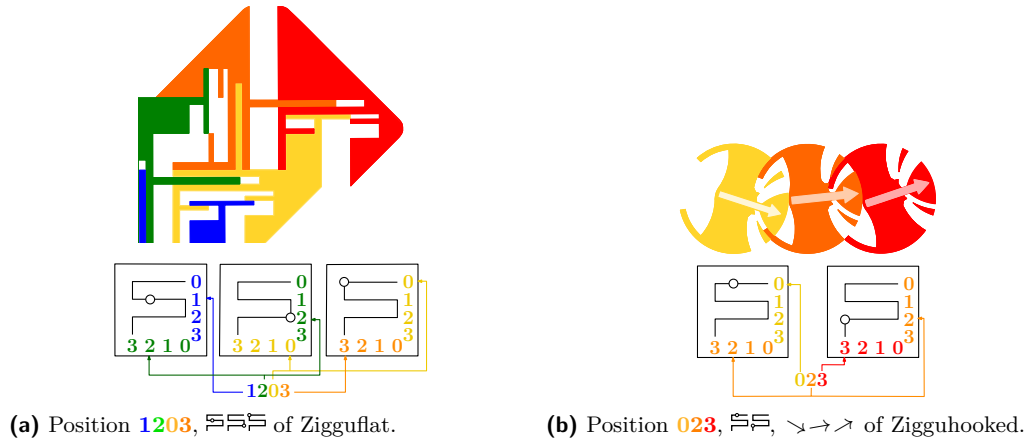
Definition 1 immediately gives a bijection between the set of m -Ziggu mazes and the set of length $m + 1$ base 4 strings in which 3 is the only digit appearing to the right of a 3.

4.2 Implementation of Ziggu mazes in the puzzles

The first puzzle in the Ziggu family was Ziggurat. Each Ziggurat piece contains a Ziggu maze, and successive pieces have alternating chirality. For example, the blue pieces in Figure 6a have an $\curvearrowright$ -shaped maze and the orange pieces have an $\curvearrowleft$ -shaped maze. The bottom of each piece contains a nub that navigates the maze two pieces below it. For example, the nub of the top blue piece passes through the top orange piece and stays within the maze of next blue piece. The puzzle is solved when every maze is solved (i.e., every nub reaches the end of its respective maze) which allows it to be disassembled. The challenge comes from the fact that consecutive pieces move in lockstep in one of the two directions of the maze. This is accomplished by the railing at the bottom of each piece slotting into the channel of the piece below it, as shown on the right side of 6a. As a result, making progress in one maze can involve moving to the opposite end of the previous maze, and this cascade of back-and-forth movements gives the puzzle its exponential nature. The Ziggurat puzzle inspired Oskar van Deventer to create the vast family of Ziggu puzzles shown in Figure 6. These puzzles all use the Ziggu maze, but they feel distinct and provide differing enjoyable puzzling experiences.

The Ziggufat puzzle consists of $p = 6$ interlocking pieces as shown in Figure 6b. The mazes are cut into the green, yellow, orange, and red pieces, and those mazes are navigated by T-shaped nubs of the purple, blue, green, and yellow pieces, respectively. Therefore Ziggufat has $m = p - 2 = 4$ mazes. Notice that the orange and red pieces only do not have T-shaped nubs and the purple and blue pieces do not have mazes so that the fully assembled puzzle is a rectangle. As the commercial version of Ziggufat has $m = 4$ mazes, it requires $n = m + 1 = 5$ digits to describe its state. See Figure 10a for an example. To read off the digits describing the state of the commercial Ziggufat puzzle with 6 pieces, the first four digits describe the vertical positions of the purple, blue, green, and yellow nubs within the mazes they navigate, respectively. The final fifth digit describes the horizontal position of the yellow nub within the red maze.

The Zigguhooked puzzle consists of p interlocking dials with $p = 6$ in the commercially available version (see Figure 6f.) Each maze is formed at the interface of two consecutive dials, so there are $m = p - 1$ mazes in total. The rotation of the dials are described by $n = p$ digits. Given a pair of adjacent dials, rotating the left dial changes the vertical position within the maze, while rotating the right dial changes the horizontal position within the maze. One can also read the digits off of the dials without examining the mazes. The radial



■ **Figure 10** The Ziggu states 1203 and 023 in Zigguflat and Zigguhooked, respectively. These are shorter versions of the puzzles with $m = 3$ and $m = 2$ mazes, respectively.

positions are numbered increasing in a counterclockwise direction, where the states 0, 1, 2, and 3 correspond to the arrow pointing in the approximately 3:45, 3:15, 2:45, and 2:15 directions of the hour hand on a clock face, respectively. So a Zigguhooked puzzle in the state 0213 would have its dials in position $\searrow \rightarrow \rightarrow \rightarrow$. See Figure 10b for an example.

5 Recurrences and Closed Formulae: Solution Lengths

In this section, we consider how many moves it takes to solve a puzzle by using recursive formulae that express the number of moves in terms of the number of moves that solve smaller versions of the puzzle. In other words, we repeat the classic approach to analyzing the Towers of Hanoi (see Figure 1) to other Gray code puzzles and Ziggu puzzles. In each case, we consider the number of moves $f(n)$ and the number of states $g(n)$ on a solution separately. These two quantities are closely related – the number of states is one more than the number of moves (i.e., $g(n) = f(n) + 1$) – but the recursive formulae arrive at these values in different ways. We also argue that these recurrences give the unique shortest (or longest) solutions. The recurrences are summarized in Tables 5–6.

5.1 Gray Code Puzzle Solution Lengths

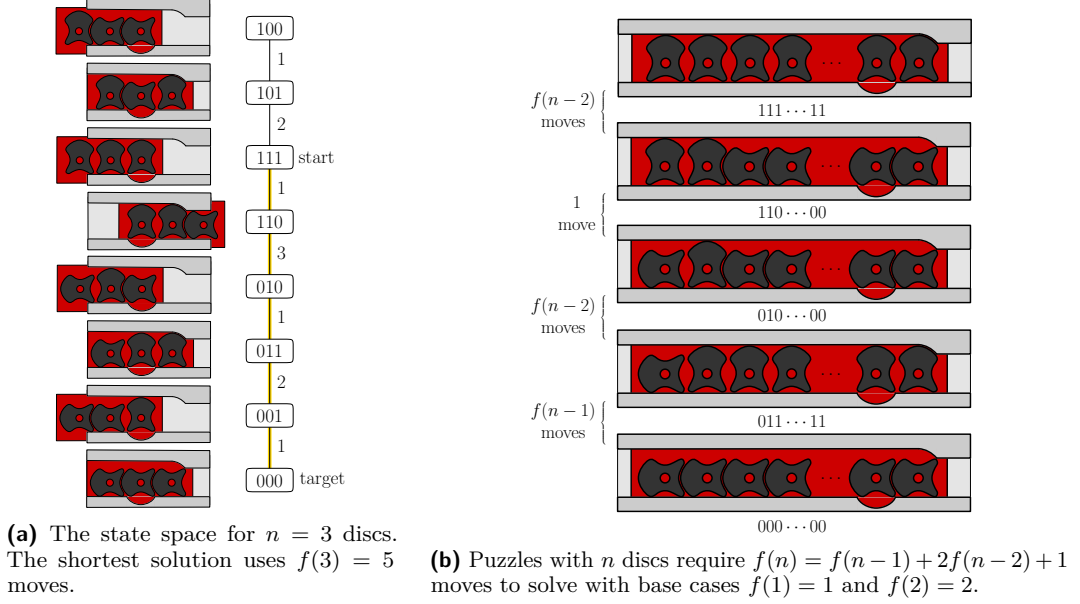
As was shown in Table 2, most Gray code puzzles run between states that are encoded as 0^n and 10^{n-1} , or between 0^n and 1^n (or vice versa).

We previously saw the recurrences for the number of moves $f(n)$ and the number of states $g(n)$ required to solve the Towers of Hanoi in Section 1. These holds for B-Bar as well. The two recurrences are provided in the top of Table 5. Theorem 2 substitutes in closed forms of these recurrences.

► **Theorem 2.** *The number of states in the shortest solution to the Towers of Hanoi with n discs is $f(n) = 2^n$. The number of moves on the shortest solution is $g(n) = 2^n - 1$. These quantities count the number of strings between 0^n and 10^{n-1} in the binary reflected Gray code.*

Recurrences for the number of moves $f(n)$ and the number of states $g(n)$ required to solve Spin-Out appear in Figure 11b. These hold for Baguenaudier and The Brain as well. The two recurrences are provided in the bottom of Table 5. Theorem 3 substitutes in closed forms of these recurrences.

► **Theorem 3.** *The number of states in the shortest solution to Spin-Out with n dials is $f(n) = \lceil \frac{2^{n+1}}{3} \rceil$. The number of moves on the shortest solution is $g(n) = \lfloor \frac{2^{n+1}}{3} \rfloor$. These quantities count the number of strings between 0^n and 1^n in the binary reflected Gray code.*



■ **Figure 11** Solving Spin-Out puzzles. An example state space is shown in (a) and the general recurrence for the number of moves required is in (b). Note that the first state is 1^n (and not 0^n) and this causes two complications: the initial move can be in the wrong direction and the sequence of dials turned $1, 3, 1, 2, 1, \dots$ is not the ruler sequence. The same recurrence for the number of moves holds for other Gray Code puzzles between 0^n and 1^n (or vice versa) including The Brain.

5.2 Ziggu Puzzle Solution Lengths

Ziggu puzzles always have the same start state and target state, so these factors do not result in different recurrences. However, these puzzles have different shortest and longest solutions, so we will still develop two different pairs of recurrences. The principle behind our recurrences can be found in Figure 12. More specifically, this figure shows how the shortest and longest solutions in a Ziggu maze puzzle can be expressed in terms of the solutions to smaller Ziggu maze puzzles. Figure 13 and 13 then illustrates the recurrences for the shortest and longest solutions in Zigguhooked, respectively. All four recurrences and summarized in Table 6. Theorem 4 substitutes in closed forms of the respective formulae.

► **Theorem 4.** *The number of states in the shortest and longest solutions to Ziggu puzzles with n digits is $f_S(n) = 6 \cdot 2^n - 3 \cdot n - 5$ and $f_V(n) = (3^{n+1} - 1)/2$, respectively. These quantities count the number of Ziggu and valid strings between 0^n and 3^n in the quaternary Gray code, respectively. The number of moves is given by $f(n) - 1$.*

6 Reflected Gray Codes: Binary and Quaternary

In this section we discuss the Gray codes that underlie Gray code puzzles and Ziggu puzzles. The term *Gray code* can broadly refer to a listing of some set of objects in which consecutive objects differ in a small amount, and in this section we'll consider Gray codes

■ **Table 5** Formulae for the unique solution for two types of Gray code puzzles with n pieces. The base cases are for $n = 1, 2$.

puzzle	quantity	recurrence	base cases	closed form	sequence
Towers of Hanoi	moves	$2f(n-1) + 1$	1	$2^n - 1$	OEIS A000225
Towers of Hanoi	states	$2g(n-1)$	2	2^n	OEIS A000079
Spin-Out	moves	$f(n-1) + 2f(n-2) + 1$	1, 2	$\lfloor 2^{n+1}/3 \rfloor$	OEIS A000975
Spin-Out	states	$g(n-1) + 2g(n-2) - 1$	2, 3	$\lceil 2^{n+1}/3 \rceil$	OEIS A005578

■ **Table 6** Formulae for the unique shortest and longest solutions to Ziggu puzzles with n digits. The base cases are for $n = 1, 2$. Note the similarity between the recurrence for the fewest moves (row 3) and $3f(n-1) - 2f(n-2)$, the recurrence for 2^n .

quantity	recurrence	base cases	closed form	sequence
most moves	$3f(n-1) + 3$	3, 12	$(3^{n+1} - 3)/2$	OEIS A029858 OEIS A123109
most states	$3g(n-1) + 1$	4, 13	$(3^{n+1} - 1)/2$	OEIS A003462
fewest moves	$3f(n-1) - 2f(n-2) + 3$	3, 12	$6 \cdot 2^n - 3 \cdot n - 6$	N/A
fewest states	$3g(n-1) - 2g(n-2) + 3$	4, 13	$6 \cdot 2^n - 3 \cdot n - 5$	OEIS A101946

for n -bit binary strings and n -digit quaternary strings. Both Gray codes use reflection, which reverses a list of strings (but not the individual strings themselves). For example, $123, 456, 789^R = 789, 456, 123$.

When ordering numbers or strings, Gray codes are not the same as standard numeric or lexicographic order. For example, in standard numeric order the 8-bit binary string 01111111 is followed by 10000000 (i.e., all n bits are flipped) whereas it is followed by 01111101 (i.e., only the second rightmost bit is flipped) in the Gray code in Section 6.

The most well-known Gray code is the *binary reflected Gray code* (BRGC) or simply the *Gray code*. It is an ordering of all 2^n binary strings of length n in which successive strings differ in only a single bit. The orders for $n = 2$ and $n = 3$ are provided below, where the overlines mark the bits that change to create the next string.

$$\text{BRGC}(2) = 0\bar{0}, \bar{0}1, 1\bar{1}, 10; \quad \text{BRGC}(3) = 00\bar{0}, 00\bar{1}, 01\bar{1}, \bar{0}10, 11\bar{0}, 1\bar{1}1, 10\bar{1}, 100. \quad (5)$$

The global structure of the binary reflected Gray code is given by the following recursive formulae.

$$\text{BRGC}(1) = 0, 1; \quad \text{BRGC}(n) = 0 \cdot \text{BRGC}(n-1), 1 \cdot \text{BRGC}(n-1)^R \quad (6)$$

In this formula, the $\cdot$ denotes concatenation, the commas denote appending one list to another, and R denotes the list in reflected order. For example, the first four words of $\text{BRGC}(3)$ are identical to $\text{BRGC}(2)$ except that 0 is prefixed to them. Similarly, the last four words of $\text{BRGC}(3)$ are identical to $\text{BRGC}(2)$ written in reflected order and with 1 prefixed. The formula (6) and its resulting order are usually attributed to Frank Gray [8], but their histories go back much further [9].

Its worth noting that the Gray code can also be created using a simple greedy algorithm [24]. Start a list with the all-zeros string 0^n (where exponentiation denotes repetition) and then repeatedly create the next string in the list from the most recently added string as follows: complement the rightmost bit that creates a new string. For example, when $n = 3$ the list will begin $000, 001, 011, 010$. To create the next string we try complementing the bits in 010 from right-to-left: $01\bar{0} = 011$ is already in the list; $0\bar{1}0 = 000$ is already in the list; $\bar{0}10 = 110$ is not in the list so it is added to the end of the list. This notion of not recreating a previous string is similar to the puzzle-solving approach from Section 3.2.

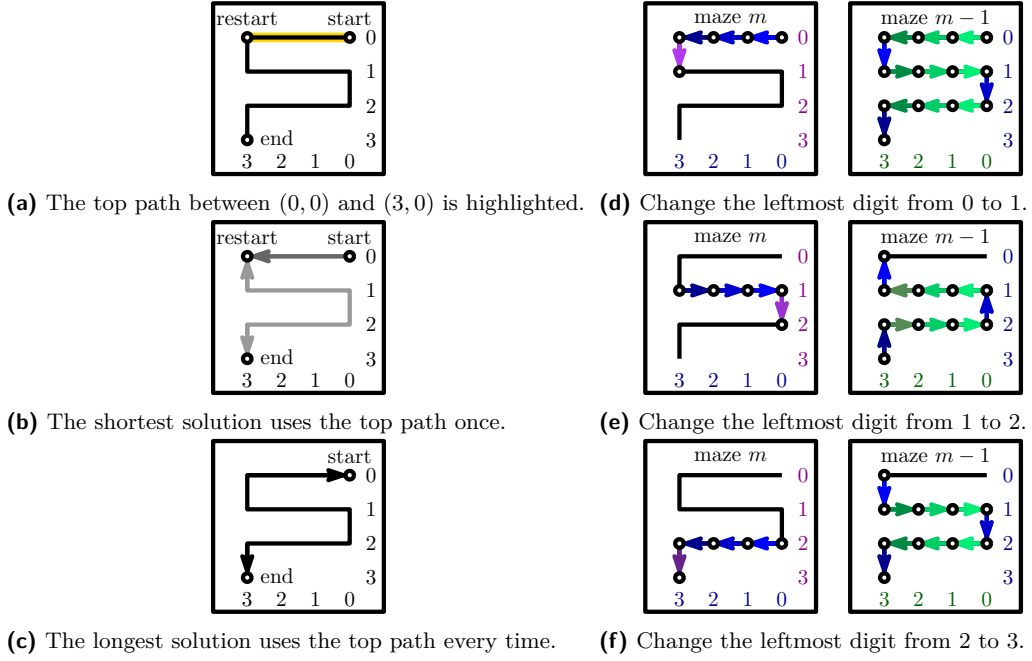


Figure 12 Solving Ziggu puzzles requires moving back-and-forth through the m mazes. However, the top path shown in (a) only needs to be traversed once. More specifically, it is traversed only the first time in (b) the shortest solution, and every time in (c) the longest solution. To solve a Ziggu puzzle we must exit the last maze (i.e., maze m). This involves changing the leftmost digit from (d) 0 to 1, then (e) 1 to 2, and finally (f) 2 to 3. The shortest solution skips the back-and-forth traversal of the top path in maze $m - 1$ in both (e) and (f). Each omission skips over all but one of the states in maze $m - 2$ (not shown).

We define the quaternary reflected Gray code recursively as follows.

$$\mathbb{Q}_1 = 0, 1, 2, 3; \quad \mathbb{Q}_n = 0 \cdot \mathbb{Q}_{n-1}, 1 \cdot \mathbb{Q}_{n-1}^R, 2 \cdot \mathbb{Q}_{n-1}, 3 \cdot \mathbb{Q}_{n-1}^R \text{ for all } n \geq 2. \quad (7)$$

Note that the global structure of the quaternary reflected Gray code is very similar to the structure of the binary reflected Gray code, as both formulas involve prepending digits to the list of one size smaller, with every second copy in reverse order.

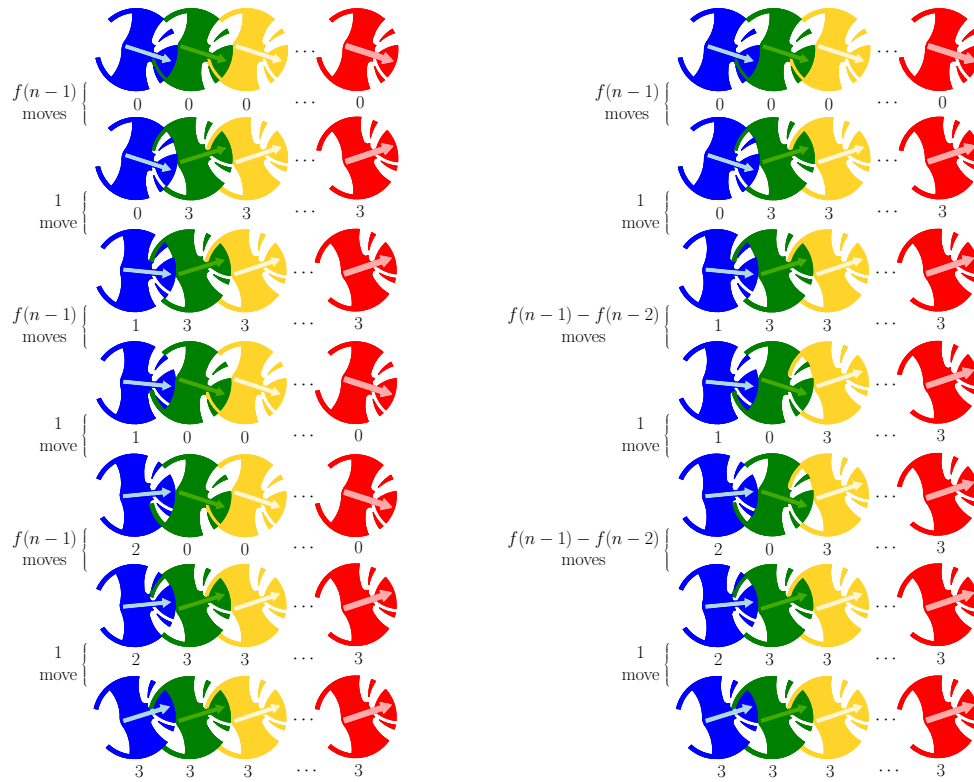
7 Recursive Formulae: The Global Structure of Solutions

Understanding how the solution of a puzzle changes as the number of mazes in the puzzle changes requires knowledge of the global structure of the solution. We describe the global structure of a sequence using a *recursive formula*, where the formula for a given n is built up from the sequences for smaller n .

The state space of many Gray code puzzles is precisely a path given by the binary reflected Gray code whose recursive description is given by (6).

The recursive structure of the longest solution to Ziggu puzzles is given by the following.

$$\mathbb{V}_1 = 0, 1, 2, 3; \quad \mathbb{V}_n = 0 \cdot \mathbb{V}_{n-1}, 1 \cdot \mathbb{V}_{n-1}^R, 2 \cdot \mathbb{V}_{n-1}, 3^n \text{ for all } n \geq 2. \quad (8)$$



(a) Longest solution. Puzzles with n discs can be solved using $f(n) = 3f(n-1) + 3$ moves. In these longest solutions no “shortcuts” are used. For example, all six optional vertices are used in Figure 4 for $n = 3$.

(b) Shortest solution. Puzzles with n discs require $f(n) = 3f(n-1) - 2f(n-2) + 3$ moves. In these shortest solutions two “shortcuts” are used and correspond to the negative terms. For example, no optional vertices are used in Figure 4 for $n = 3$.

■ **Figure 13** Recurrences for the number of moves in the (a) longest and (b) shortest solutions to Zigguhooked. Base cases are $f(1) = 3$ and $f(2) = 12$ as the state spaces for $n = 1$ and $n = 2$ are paths (see Figure 4). The recurrences apply for other Ziggu puzzles with the same number of mazes.

This recursive formula is derived from the recursive formula for the quaternary reflected Gray code, with the modification that instead of recursively including the entire list $3 \cdot \mathbb{Q}_{n-1}^R$, we append the state 3^n , as there is no need to continue beyond the solved state 3^n . See Figures 14 and 15.

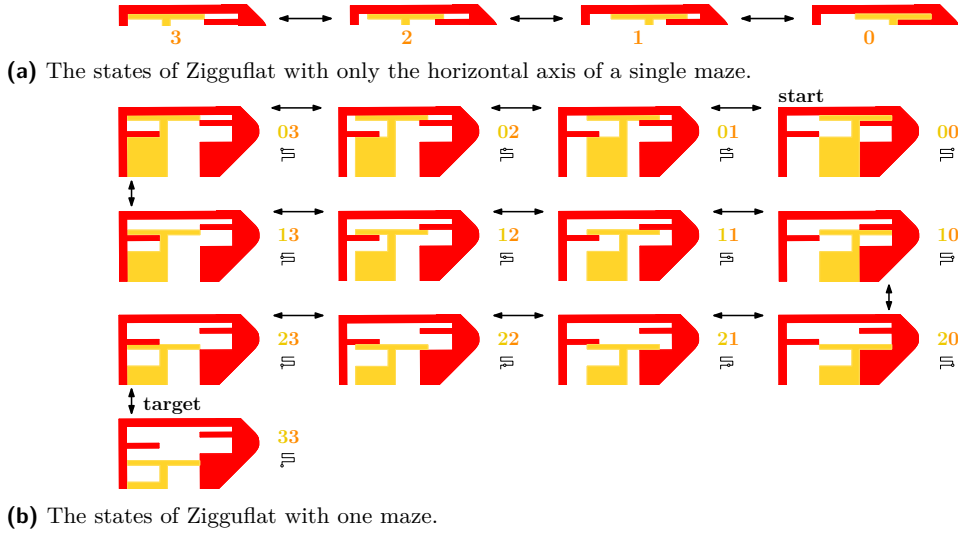
The recursive structure of the shortest solution to Ziggu puzzles is slightly more complicated. Let $\mathbb{S}_1 = \text{core}(1) = 0, 1, 2, 3$, and for $n \geq 2$, let

$$\mathbb{S}_n = 0 \cdot \mathbb{S}_{n-1}, \text{core}(n)[1:] \quad \text{and} \quad \text{core}(n) = 03^{n-1}, 1 \cdot \text{core}(n-1)^R, 2 \cdot \text{core}(n-1), 3^n. \quad (9)$$

The $\text{core}(n)[1:]$ denotes all but the first element of $\text{core}(n)$. This “list slicing” is needed because the word 03^{n-1} appears as both the last word of $0 \cdot \mathbb{S}_{n-1}$ and the first word of $\text{core}(n)$. $\mathbb{S}_n$ is the sublist of $\mathbb{V}_n$ obtained by omitting all states where 03 is followed by a digit that is not 3 .

8 Successor and Predecessor Rules: The Next and Previous States

Puzzles such as the Gray code and Ziggu puzzles can take a long time to solve, and so it is conceivable that a puzzle solver might set a puzzle down during a solving session to say, make a cup of tea. If the puzzle solver noted their last move or could recall it, they may



■ **Figure 14** This figure illustrates the recursive structure of both the longest and shortest solution to the Ziggu puzzles (they agree when $n = 1, 2$) using a version of Zigguflat with one maze. Figure 14a is the solution with only the horizontal part of the maze, which has 4 states. The successive rows in Figure 14b are created by prepending either a 0, 1, or 2 in each successive row to the state given by the column.

be able to resume their solution using the approach outlined in Section 3.2. However, if a puzzle solver has no knowledge of the previous move, they would use a *successor rule* to determine the next state to which they should advance the puzzle. Similarly, a *predecessor rule* determines the previous state in the list.

Let $2k$ be an even base (binary is $k = 1$, quaternary is $k = 2$), and let $w = w_n \cdots w_2 w_1$ be a base- $2k$ string of length n . Its *reflected Gray code successor* is obtained by incrementing or decrementing digit w_i as follows

$$w = \begin{cases} w_n \cdots w_{i+1} w_i + 1 w_{i-1} \cdots w_1 & \text{if } w_i < 2k - 1 \text{ and } \sum_{j=i+1}^n w_j \text{ is even} \\ w_n \cdots w_{i-1} w_i - 1 w_{i-1} \cdots w_1 & \text{if } w_i > 0 \text{ and } \sum_{j=i+1}^n w_j \text{ is odd} \end{cases} \quad (10)$$

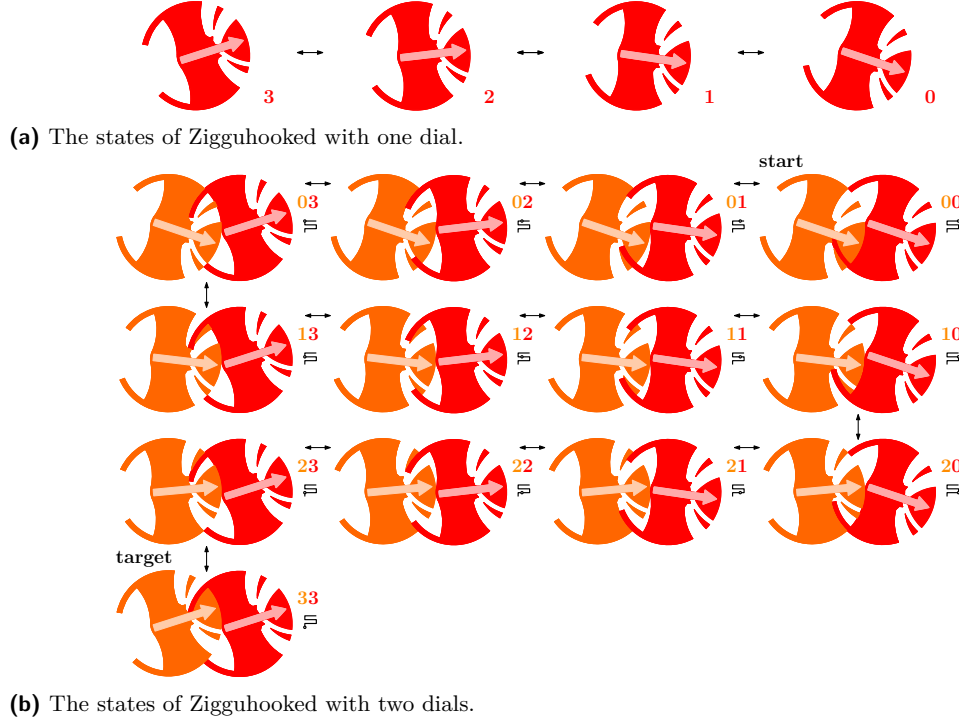
where i is the minimum index such that one of the two cases holds. If there is no such value of i , then w 's successor q is undefined.

For the quaternary ($k = 2$) string $w = 012310$, we see that $i = 2$ since $w_2 = 1 < 3$ and $\sum_{j=i+1}^n w_j = 3 + 2 + 1 + 0 = 6$ is even. (10) gives that $w = 012320$, so digit w_2 was incremented.

If w is undefined for a quaternary string $w \in \mathbb{Q}_w$, then $w = 30^{n-1}$, which is the last string in the quaternary Gray code

We now provide explicit successor rules for the longest and shortest solutions to Ziggu puzzles. These are sublists of the quaternary reflected Gray code, so we work over base 4, no longer over an arbitrary even base $2k$. Let $q = q_n \cdots q_2 q_1 \in \mathbb{V}_n$ be a long quaternary string of length n . Its *long successor* is obtained by incrementing or decrementing digit q_i as follows

$$\text{nextV}(q) = \begin{cases} q_n \cdots q_{i+1} q_i + 1 q_{i-1} \cdots q_1 & \text{if } \sum_{j=i+1}^n q_j \text{ is even and } q_i < 3 \\ q_n \cdots q_{i-1} q_i - 1 q_{i-1} \cdots q_1 & \text{if } \sum_{j=i+1}^n q_j \text{ is odd, } q_i > 0 \text{ and } q_{i-1} \neq 3 \end{cases} \quad (11)$$



■ **Figure 15** This figure illustrates the recursive structure of both the longest and shortest solution to a version of Zigguhooked with only 2 dials. Figure 15a is the solution with only 1 dial, which has 4 states. The successive rows in Figure 15b are created by prepending either a 0, 1, or 2 in each successive row to the state given by the column.

where i is the minimum index such that one of the two cases holds. If there is no such value of i , then q 's long successor $\text{nextV}(q)$ is undefined.

For example, if $q = 20103$, then $i = 1$, since $q_1 = 3 > 0$ and $\sum_{j=i+1}^n q_j = 0 + 1 + 0 + 2 = 3$ is odd. Therefore $\text{nextV}(q) = 20102$, as digit q_1 was decremented.

Comparing (10) and (11), we see that the longest solution omits the strings in the quaternary reflected Gray code where a 3 is followed by digits that are not 3.

The successor rule for the shortest solutions adds an extra condition. Let $q = q_n \cdots q_2 q_1 \in \mathbb{S}_n$ be a short quaternary string of length n . Its *short successor* is obtained by incrementing or decrementing digit q_i as follows

$$\text{nextS}(q) = \begin{cases} q_n \cdots q_{i+1} q_{i+1} q_{i-1} \cdots q_1 & \text{if } \sum_{j=i+1}^n q_j \text{ is even, } q_i < 3 \\ q_n \cdots q_{i-1} q_{i-1} q_{i-1} \cdots q_1 & \text{if } \sum_{j=i+1}^n q_j \text{ is odd, } q_i > 0, q_{i-1} \neq 3, q_{i-1}q_i \neq 03 \end{cases} \quad (12)$$

where i is the minimum index such that one of the two cases holds. If there is no such value of i , then q 's short successor $\text{nextS}(q)$ is undefined.

For example, let $q = 20103$. We see that now $i \neq 1$ since $q_2 q_1 = 03$. Instead, $i = 3$ since $q_3 = 1 < 3$ and $\sum_{j=i+1}^n q_j = 0 + 2 = 2$ is even. Therefore $\text{nextS}(q) = 20203$, where we incremented digit q_3 .

Comparing (11) and (12), we see that the shortest solution omits the strings in the longest solution where a 03 is followed by digits that are not 3.

If $\text{next}\mathbb{V}(q)$ or $\text{next}\mathbb{S}(q)$ is undefined for $q \in \mathbb{V}_n$ or $q \in \mathbb{S}_n$, then $q = 3^n$, which is the last string in both the longest and shortest solution.

9 Comparison: The Relative Position of States

Determining which moves will advance a puzzle in an arbitrary state towards the initial or solved state can seem quite daunting. In Gray code puzzles where the initial state is 1^n but the puzzle can have different valid states that also have first digit 1, one may have to “backtrack” in the Gray code to find the shortest path to the official starting state of the puzzle. A comparison formula allows a puzzle solver to determine the relative order of the current state and the solved state, thus indicating whether or not to proceed forwards or backwards in the Gray code.

For the Ziggu puzzles where the initial state 0^n and the solved state 3^n are the first and final valid states in both the longest and shortest solutions $\mathbb{V}_n$ and $\mathbb{S}_n$, a comparison formula is more generally useful when one is curious about the relative order of two puzzle states.

We present a comparison formula that can compute the relative order of two states in all four Gray codes we’re considering: the binary and quaternary reflected Gray codes, and the longest and shortest solutions to the Ziggu puzzles.

Given two distinct states $w, v \in \text{BRGC}(n)$ with $w = w_n \cdots w_2 w_1$ and $v = v_n \cdots v_2 v_1$, we can determine if w occurs before v in the binary reflected Gray code as follows. Let w_k be the leftmost digit where w and v differ, i.e, k is the index such that $w_i = v_i$ for all $k < i \leq n$ and $w_k \neq v_k$. We compute that w occurs before v as follows.

$$w < v \text{ if } \begin{cases} w_k < v_k & \text{if } \sum_{i=k+1}^n w_i \text{ is even} \\ w_k > v_k & \text{if } \sum_{i=k+1}^n w_i \text{ is odd.} \end{cases} \quad (13)$$

For example, $101 < 100$ in $\text{BRGC}(3)$, since $k = 1$, $w_3 + w_2 = 1 + 0$ is odd, and $1 > 0$. The parity condition on the sum of the digits to the left of w_k captures the number of times that the recursive sublists are written in reverse order.

The comparison in (13) also gives the relative order for distinct states w and v in the quaternary reflect Gray code $\mathbb{Q}_n$, as the parity condition still captures this same sublist reversal behavior. For example, $11013 < 11023$ in $\mathbb{Q}_5$ since $k = 2$, $w_3 + w_4 + w_5 = 2$ is even, and $1 < 2$.

Since both the longest and shortest solutions are sublists of $\mathbb{Q}_n$, we see immediately that if we are given distinct states $w, v \in \mathbb{V}_n$ or $w, v \in \mathbb{S}_n$, then (13) provides their relative order within these respective lists.

10 Loopless Algorithms for Ziggu Puzzles

Combinatorial generation refers to the efficient generation of every object of a particular type [18]. In this context, the best type of algorithm is a *loopless algorithm*, which creates each successive object in worst-case $O(1)$ -time. More specifically, a single object is stored in memory, and the generation algorithm makes a constant-sized modification to obtain the next object. Loopless algorithms were pioneered by Ehrlich [4] and exist for most common types of combinatorial objects (e.g., all n -bit binary strings, all permutations of $\{1, 2, \dots, n\}$, etc.). In this section we provide loopless algorithms for the shortest solutions to Ziggu puzzles.

We present a loopless algorithm for generating the shortest solution to Ziggu puzzles, provided one stores a “direction vector,” in Algorithm 1. Algorithm 1 has the following initial state: initial word $w = 0^n$ and starting index $i = 1$. The array of direction d is initialized to have every entry be a 1, indicating that each digit should be increased (lines 2-4).

In every loop iteration, Algorithm 1 changes w_i by the value $d[i]$, the i -th right-to-left 1-indexed value in the array d (line 7). If this change results in w_i being extremal, so w_i equals 0 or 3, $d[i]$ is set from increasing (1) to decreasing (-1) or vice versa (line 9). In the next loop iteration, the digit to the left of w_i will be changed (line 10). If w_i was instead changed to a nonextremal value (1 or 2), then the next loop iteration will consider the digit to the right of w_i , if w_i is not the rightmost digit (lines 11-12). If w_i is the rightmost digit, w_i will be changed again in the next loop iteration.

We remark that the memory required for this algorithm can be reduced, as the storage of the directions is not strictly necessary. However, storing the direction vector simplifies the algorithm.

■ **Algorithm 1** An algorithm for generating the shortest solution to the Ziggu puzzles. Each successive state is generated in worst-case $O(1)$ time, making it a loopless algorithm. The algorithm uses one index variable, which requires $O(\log(n))$ memory, and n bits of additional memory to store directions.

```

1: procedure FASTZIGGUWITHDIRECTIONS( $n$ )
2:    $d \leftarrow [1] * n$                                 ▷ The starting directions, a length  $n$  array of 1s.
3:    $w \leftarrow 0^n$                                     ▷ The starting word of  $n$  zeroes.
4:    $i \leftarrow 1$                                        ▷ 1-based (right-to-left) index for changing  $w$ .
5:   visit  $w$ , None
6:   while  $i \leq n$  do
7:      $w_i \leftarrow w_i + d[i]$                           ▷ Change  $w_i$  by  $\pm 1$ , depending on value of  $d[i]$ .
8:     if  $w_i \in \{0, 3\}$  then                            ▷ Check if  $w_i$  is now extremal.
9:        $d[i] \leftarrow -d[i]$                              ▷ Change  $d[i]$  from  $\pm 1$  to  $\mp 1$ .
10:     $i \leftarrow i + 1$                                    ▷ Consider the letter to the left.
11:    else if  $i > 1$  then
12:       $i \leftarrow i - 1$                                 ▷ Consider the letter to the right.
13:    visit  $w$ ,  $i$ 

```

11 Final Remarks

Preliminary results in this paper were shared by the second author at a *Gathering for Gardner – Celebration of Mind* presentation. This led to several enjoyable communications with the community including Bram Cohen, Oskar van Deventer, and Goetz Schwandtner. Along with Hirokazu Iwasawa they have been making parallel investigations into Ziggu puzzles and their generalizations. In particular, they have considered variations to the basic S-shaped maze and how this changes the solution lengths. This research has resulted in new puzzle designs, including van Deventer’s recent Zigguphi puzzle. The authors look forward to playing these new puzzles, and hope that some of the results presented here will help in their analysis. For those who would like to try their hand at a physical copy of these puzzles, please consult van Deventer’s webpage of freely available 3D-printing files [23] or visit an online puzzle retailer.

The authors of this paper have noticed that our results generalize to mazes that continue the back-and-forth pattern (i.e., $2g + 1$ rows with $g \geq 1$). For further technical results, including change sequences, ranking algorithms, an additional algorithm for generating the shortest solution, and a bijection with the pencil-and-paper puzzle *Nurikabe*, see the preliminary version [7] of this paper.

References

- 1 Puzzle Design Competition Committee. 2021 Puzzle Design Competition Entries. URL: <https://johnrausch.com/DesignCompetition/2021/>.
- 2 Curtis Cooper. The Chinese ring puzzle, the crazy elephant dance puzzle, the b-spinout puzzle, and Gray codes. *The Mathematical Gazette*, 103(558):431–441, 2019.
- 3 Paul Cull, Leanne Merrill, and Tony Van. A tale of two puzzles: Towers of Hanoi and Spin-Out. *Information and Media Technologies*, 8(3):695–709, 2013. doi:10.11185/IMT.8.695.
- 4 Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. ACM*, 20(3):500–513, 1973. doi:10.1145/321765.321781.
- 5 Martin Gardner. Curious properties of the Gray code and how it can be used to solve puzzles. *Scientific American*, 227(2):106–109, 1972.
- 6 Martin Gardner. *Knotted doughnuts and other mathematical entertainments*. AMS, 2020.
- 7 Madeleine Goertz and Aaron Williams. The quaternary Gray code and how it can be used to solve Ziggurat and other Ziggu puzzles. *arXiv preprint arXiv:2411.19291*, 2024.
- 8 Frank Gray. Pulse code communication, March 1953.
- 9 FG Heath. Origins of the binary code. *Scientific American*, 227(2):76–83, 1972.
- 10 Felix Herter and Günter Rote. Loopless Gray code enumeration and the Tower of Bucharest. *Theoret. Comput. Sci.*, 748:40–54, 2018. doi:10.1016/J.TCS.2017.11.017.
- 11 Andreas M Hinz, Sandi Klavžar, Uroš Milutinović, and Ciril Petr. *The Tower of Hanoi-Myths and Maths*. Birkhäuser, 2nd edition, 2018.
- 12 Donald E Knuth. *The art of computer programming, volume 4A: combinatorial algorithms, part 1*. Pearson Education India, 2011.
- 13 Torsten Mütze. Combinatorial Gray codes - an updated survey, 2023. arXiv:2202.01280.
- 14 Torsten Mütze. On Hamilton cycles in graphs defined by intersecting set systems. *Notices Amer. Math. Soc.*, 71(5), 2023.
- 15 Ed Pegg Jr. The icosian game, revisited. *The Mathematica Journal*, 11(3):310–314, 2009.
- 16 Kirk Pruhs. The Spin-Out puzzle. *ACM SIGCSE Bulletin*, 25(3):36–38, 1993. doi:10.1145/165408.165421.
- 17 Lisa Rougetet. *Le binaire au bout des doigts: Un casse-tête entre récréation mathématique et enseignement*. UGA Éditions, 2024.
- 18 Frank Ruskey. Combinatorial generation. *Preliminary working draft. University of Victoria, Victoria, BC, Canada*, 11:20, 2003.
- 19 Goetz Schwandtner. *n*-ary puzzles, November 2025. URL: https://puzzles.schwandtner.info/group_nary.html.
- 20 RS Scorer, Patrick Michael Grundy, and Cedric Austen Bardell Smith. Some binary games. *The Mathematical Gazette*, 28(280):96–103, 1944.
- 21 Paul K Stockmeyer. An exploration of sequence A000975. *Fibonacci Quart.*, 55(5):174–185, 2017.
- 22 Jim Storer. Spinout. URL: <https://www.cs.brandeis.edu/~storer/JimPuzzles/ZPAGES/zzzSpinout.html>.
- 23 Oskar van Deventer. Print it yourself. <https://oskarvandeventer.nl/Print-It-Yourself/>, 2022.
- 24 Aaron Williams. The Greedy Gray Code Algorithm. In Frank Dehne, Roberto Solis-Oba, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, pages 525–536, Berlin, Heidelberg, 2013. Springer. doi:10.1007/978-3-642-40104-6_46.