Permutation Puzzles

Mu Alpha Theta

Aditya Pahuja

Taught on May 20, 2024

Contents

1 The 15-Puzzle

The **15-Puzzle** is a classic children's puzzle in which you are given a fifteen tiles numbered 1 through 15, (somewhat) arbitrarily placed on a 4×4 grid; you are allowed to slide the tiles around via the one empty space, and your goal is to restore the board to the state where the tiles are in increasing order (pictured below).

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

We are of course not constrained to just 4×4 grids — any $m \times n$ grid with tiles labeled 1, 2, ..., $mn - 1$ (where m and n are at least 2) would give a variant of the puzzle.

Puzzles like this are called **permutation puzzles**, since you can identify each element of the puzzle with a number and the moves you can make correspond to permuting the numbers. The goal of any permutation puzzle is to move the elements around to achieve a particular "solved state." In the 15-Puzzle, it'll be helpful to pretend that the blank space is labeled 16, so that we can describe the state of the grid with a permutation of 1, 2, ..., 16 (however, we will assume that the bottom-right square always starts empty because we can always restore the board to a configuration satisfying that). The moves we can make are the swaps involving 16 (e.g. we can swap 15 and 16 or 12 and 16 in the picture above), and the solved state is when all of 1 through 16 are in order when read left to right, top to bottom.

A natural question that appears in the theory of permutation puzzles is what kinds of permutations are "solvable" — if we were to place down the pieces in the puzzle at random, how would we be able to figure out whether or not the puzzle can be returned to the solved state? Historically, this has shown up in culture as the question "Is the following board solvable?"

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

1.1 Experimentation: DO THIS BEFORE READING ON

To answer the question of when the puzzle is solvable, we need to understand its structure, so it is crucial to familiarize yourself with how it works. Thus, you should playing it for a while; an online version can be found here.

Problem 1. *Describe a general strategy to solve the puzzle for any $m \times n$ grid, with $m, n \geq 2$, or at least some guidelines/heuristics for how you can get yourself "reasonably close" to winning. The goal we are working towards here is to articulate an algorithm that a perfect follower of instructions (e.g. a computer) could follow to solve the puzzle, but for now just getting an intuitive sense of what works is a good start. Don't worry about optimizing how many moves you spend.*

At minimum, though, you should come up with an algorithm to solve 2×3 grids, since that will help with the next problem.

Problem 2. *For the opposite direction, we want to get a handle on what kinds of tile positions might cause obstructions to winning. Play with the following boards and decide whether or not they're solvable (either by solving them or convincing yourself that it's impossible):*

1	2	3
5	4	

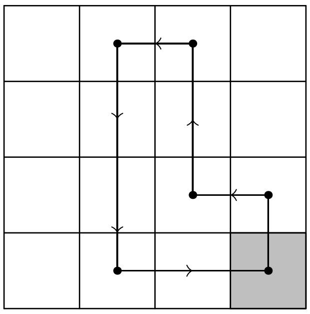
2	1	3
5	4	

3	1	2
4	5	

Try some other initial positions of a 2×3 *grid and see if you can notice any patterns.*

1.2 What do solvable grids look like?

Let's suppose that some permutation of 1, 2, ..., mn results in a solvable configuration of an $m \times n$ grid. We can assume that the blank square is at the bottom-right corner, so in the starting position the last element of the permutation is mn. Then, since the board is solvable, there is some sequence of swaps s_1, s_2, ..., s_k that restores the board to the solved state, and importantly puts the blank space back at the bottom-right corner.



The main observation here is that k, the number of swaps, will always be even. This actually gives us some information about the permutation — that is, not every permutation can be obtained via an even number of swaps. (Note that "swaps" refers to switching any two elements in the permutation, not just two adjacent ones.)

Proposition 1.1. Say a permutation of 1, 2, ..., N is **even** if it can be obtained via applying an even number of swaps to $(1, 2, \ldots, N)$ and **odd** otherwise. Then:

(i) Even permutations correspond to the permutations $(a_1, a_2, \ldots, a_N)$ in which the number of pairs that are out of order (i.e. the number of pairs (a_i, a_j) such that $i > j$ but $a_i < a_j$) is even.

(ii) The number of even permutations is the same as the number of odd permutations.

Proof. Let's start by proving (i).

Suppose first that the number of flipped pairs is even. Then, we want to show that the number of swaps needed to restore the permutation to $(1, 2, \ldots, N)$ is also even, since we can reverse the order of the swaps to go from $(1, 2, \ldots, N)$ to $(a_1, a_2, \ldots, a_N)$ with an even number of swaps. Zooming in locally at one swap, we have something like

$$(a_1, a_2, \ldots, a_i, a_{i+1} \ldots, a_j, \ldots, a_N) \mapsto (a_1, a_2, \ldots, a_j, a_{i+1}, \ldots, a_i \ldots, a_N).$$

This results in flipping the order of each of (a_i, a_t) and (a_j, a_t) for $i + 1 \leq t \leq j - 1$, as well as (a_i, a_j). Each of these pairs changes whether or not they're flipped, which, because the number of altered pairs is odd, shifts the total count of flipped pairs by an odd number. In other words, *swaps change the parity of the number of flipped pairs.* Then, the starting number of flipped pairs is even and so is the ending number (zero), meaning an even number of swaps had to be made.

Conversely, if the number of swaps made is even, then the number of flipped pairs will be even by the exact same logic.

For (ii), if E is the set of even functions and O the set of odd functions, we can construct a simple bijection $E \to O$ by

$$(a_1, a_2, \ldots, a_N) \mapsto (a_2, a_1, \ldots, a_N).$$

We know this is a bijection because the function has an obvious inverse (by just swapping the first two elements back), so the sizes of E and O have to be the same. □

This proposition is great because it

1. shows us that "even permutation" is actually a meaningful designation in that there exist non-even permutations (it would be a silly label if all permutations were even), and

2. gives us a simple criteria for deciding whether a particular permutation is even.

In fact, we can immediately say now that, since $(1, 2, 3, 5, 4, 6)$ is an odd permutation in which 6 is the last element, the first board of Problem 2 is unsolvable!

You should have found that the other two boards, which are given by even permutations, are solvable; we would like to assert that this is true in general, i.e. all boards given by even permutations are solvable. Happily enough, this is true, and to prove this, we need to develop a general strategy for winning this game.

We will begin this with a perspective shift, though: Instead of thinking of a permutation as a list $(a_1, a_2, \ldots, a_N)$, we will instead talk about it as a bijective function $\pi \colon \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ with $\pi(i) = a_i$. In other words, permutations are actions on a list of objects rather than particular lists of numbers.

Definition 1.2 (Cycle notation) — A **cycle**, denoted by $(a_1\ a_2\ \cdots\ a_r)$, represents the permutation

$$a_1 \mapsto a_2 \mapsto \cdots \mapsto a_{r-1} \mapsto a_r \mapsto a_1,$$

i.e. the permutation that sends the object at position a_i to position a_{i+1} (note that $a_{r+1} = a_1$ and all other objects are fixed by this permutation). The **length** of this cycle is r.

To emphasize: cycles are just permutations, so, just like we can compose two permutations $\pi_1 \circ \pi_2$, we can compose two cycles, which we denote by just writing the cycles next to each other, like

$$(a_1\ a_2\ \cdots\ a_r)(b_1\ b_2\ \cdots\ b_s).$$

Notably, composition of permutations will be *left-to-right*, i.e. $(\pi_1 \circ \pi_2)(x) = \pi_2(\pi_1(x))$. For example, we can represent any permutation by a sequence of swaps

$$(a_1\ b_1)(a_2\ b_2) \cdots (a_r\ b_r).$$

Proposition 1.3. A permutation is even if and only if it can be written as $C_1 \circ C_2 \circ \cdots \circ C_r$, where each of the C_i is a 3-cycle (cycle of length 3).

Proof. First, we see that 3-cycles are even because $(x\ y\ z) = (y\ z)(x\ y)$, so, if the C_i are all 3-cycles, then we can unfurl $C_1 \circ C_2 \circ \cdots \circ C_r$ into a bunch of pairs of swaps.

Conversely, if a permutation is even, then it can be written as

$$[(a_1\ b_1)(c_1\ d_1)][(a_2\ b_2)(c_2\ d_2)] \cdots [(a_r\ b_r)(c_r\ d_r)],$$

the composition of a bunch of pairs of swaps. Then, in general, we would like to write pairs of swaps $(w\ x)(y\ z)$ as compositions of 3-cycles. Obviously $w \neq x$, $y \neq z$. We then have three cases:

- If $\{w, x\} = \{y, z\}$, then the two swaps act on the exact same pair of objects and so they cancel each other out; this is trivially a composition of zero 3-cycles.

- If one of w, x is equal to one of y, z — say WLOG $w = z$ — then

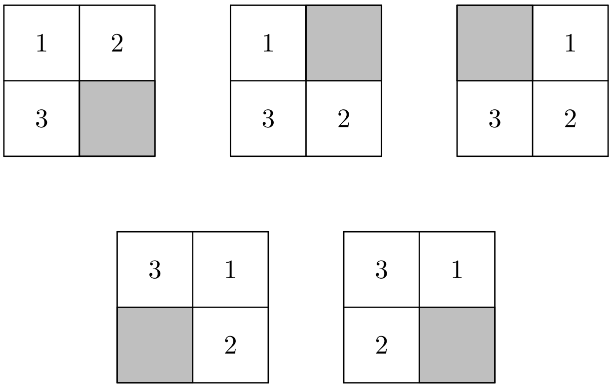
$$(w\ x)(x\ y) = (y\ x\ w).$$

- If w, x, y, and z are all distinct, then write

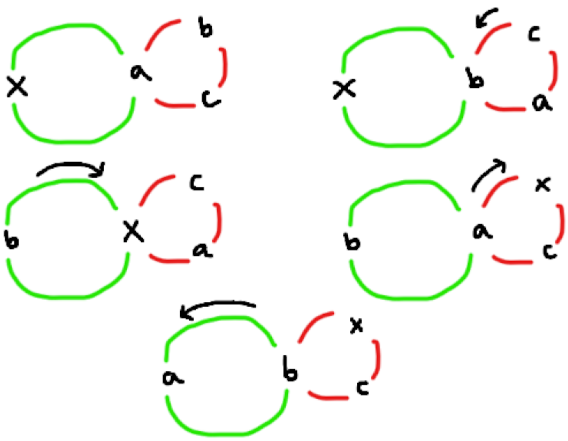
$$(w\ x)(y\ z) = (w\ x)[(x\ y)(x\ y)](y\ z) = [(w\ x)(x\ y)][(x\ y)(y\ z)] = (y\ x\ w)(z\ y\ x).$$

This means each $(a_i\ b_i)(c_i\ d_i)$ can be expressed as a composition of 3-cycles, so any even permutation is a composition of 3-cycles. $\square$

This is helpful because it rephrases even permutations in a more tractable form: as it turns out, doing arbitrary 3-cycles is a lot easier than trying to perform arbitrary pairs of swaps. In fact, 3-cycles are actually a pretty natural operation in the confines of the puzzle — here's a very simple one right here corresponding to $(1\ 2\ 3)$.



This lends itself to a trick: we can create other 3-cycles by swapping pieces into this 2×2 square mid-cycle. Here's an abstract picture of what we're going to do, extending the cycle $(c\ b\ a)$ into a cycle $(x\ b\ a)$:



Problem 3. *What does this look like in practice? More explicitly: Figure out how to do the 3-cycle* $(a\ 11\ 12)$ *in the* 4×4 *version of the puzzle, where* a *is any positive integer between* 1 *and* 15 *except* 11 *or* 12.

Once we have the result of Problem 3, we can obviously also do the 3-cycle $(a\ 12\ 11)$ by applying $(a\ 11\ 12)$ twice. This is all the machinery we need to do an arbitrary 3-cycle $(a\ b\ c)$ in the case where a, b, c are all distinct from 11, 12:

	a		
b			c
		11	12

	12		
b			c
		a	11

	12		
a			c
		11	b

	12		
a			b
		c	11

	c		
a			b
		11	12

In cycle notation,

$$(a\ b\ c) = (a\ 11\ 12)(b\ 12\ 11)(c\ 11\ 12)(a\ 12\ 11).$$

Problem 4. *How do you do the 3-cycle $(a\ b\ 11)$, where a and b are not 11 or 12?*

This means we can do *any* 3-cycle, so we have come up with an algorithm to solve the puzzle!

To finish off, let's also look at what happens if the empty square is allowed to be at any location in the initial configuration of the grid. We know how the game plays out once the empty square is at the bottom-right cell, so let's consider what moving the empty square to the corner looks like. Each swap will reverse the parity of the permutation corresponding to the current board state, which means that

- if the permutation is odd, the number of moves needed to move to the corner is also odd;
- if the permutation is even, the number of moves needed to move to the corner is also even.

More neatly, we can say that the **taxicab distance** between the empty cell and the bottom right corner, which is the sum of the horizontal and vertical distances between the empty cell and the corner, always has the same parity as the permutation corresponding to the board state. In conclusion:

Theorem 1.4. A given board state of the 15-Puzzle is solvable if and only if the corresponding permutation has the same parity as the taxicab distance between the empty cell and the bottom-right corner.

Problem 5. *How many solvable board states are there?*

2 Abstraction, and a harder puzzle

Before going on to talk about a much more famous permutation puzzle, I want to touch a little bit on the underlying theory that happened in the process of solving the 15-Puzzle. The key part of the solution algorithm was the ability to extend a specific 3-cycle into a more general 3-cycle. This extension had four parts:

(1) partially doing a 3-cycle in the bottom-right corner, say via a sequence of moves C;

(2) substituting one of the numbers being cycled with an arbitrary piece in the grid, say via a sequence of moves S;

(3) undoing the 3-cycle; and

(4) undoing S.

Generally we represent this sequence of operations by $CSC^{-1}S^{-1}$. This is called a **commutator** and is denoted $[C, S] = CSC^{-1}S^{-1}$. Commutators are the bread and butter of permutation puzzles, since, if the set of pieces moved by C shares very few pieces with the set of pieces moved by S, then $[C, S]$ lets you move around a very small set of pieces (this is exactly what happened when we built out the more general 3-cycle).

Remark 2.1 (Commutators in group theory)**.** The name "commutator" arises from the following idea: let's say you have a group G, and you want to understand its abelian (commutative) subgroups. We can try getting at the subgroups by looking at homomorphisms from G to some abelian group A, say $\varphi\colon G \to A$. Since A is abelian, we need

$$\varphi(gh) = \varphi(g)\varphi(h) = \varphi(h)\varphi(g) = \varphi(hg),$$

or equivalently $\varphi(ghg^{-1}h^{-1}) = \varphi([g, h]) = 1$ for every g and h in G. This means every single commutator of elements in G is in the kernel of φ (i.e. the set of things that get sent to the identity by φ), so abelian subgroups of G turn out to correspond to normal subgroups of G that contain every commutator in G.

2.1 Rubik's cube

This abstraction is useful when dealing with a Rubik's cube, which has a lot more complexity. We will do a similar analysis as we did with the 15-Puzzle, where we look at the set of positions that can be obtained from taking apart a cube and putting it back together and we want to find which ones can, via some sequence of turns, reach the solved state of the cube. I will, however, leave the details (interesting parts) as problems.

While a Rubik's cube is a permutation puzzle in the sense that we can label the 54 stickers and then consider the configuration as a permutation of the numbers, this mushes together a lot of the structure embedded in the puzzle. Rather, we can consider the edge and corner pieces separately.

The canonical way to understand the position of a Rubik's cube is to look at the locations of the pieces and their "orientations" separately (in particular, the former isn't enough information; for example, it is possible to reach the position where every corner is solved and every edge is in the correct spot but flipped).

It will help to read on while having a Rubik's cube on hand.

Definition 2.2 (Orientation) — Suppose the cube is held in "standard position," i.e. the white center is on top (the U face) and the green center is facing you (the F face). Then, to each edge, we assign the number 0 if it can be moved to its home location, with the correct orientation, without rotating the front or back (blue or green) faces; otherwise, we assign it the number 1. To each corner, we assign the smallest number of clockwise twists necessary so that its white/yellow sticker faces up/down (whichever applies).

Problem 6. *Suppose you are given a solvable configuration of a cube. Show that the sum of the edge numbers is even and the sum of the corner numbers is odd.*

In some sense, this definition of orientation is kind of arbitrary; the important part is that it embeds the idea that, if we were to reassemble a Rubik's cube, the orientations of the first 11 edges and the first 7 corners that you place will fix the orientation of the last edge and corner. In other words, if you have ever twisted a corner of a friend's cube and then asked them to solve said cube, you are a menace for giving them an unsolvable cube.

Separately, we can also look at how the pieces themselves are moving, without regard to their orientation.

Problem 7. *The locations of the edges and corners, ignoring orientation, induce permutations of* $\{1, 2, \ldots, 12\}$ *and* $\{1, 2, \ldots, 8\}$*. Show that, in a solvable configuration, the edge permutation and the corner permutation will have the same parity.*

It turns out that this is the most we can constrain solvable positions:

Problem 8. *Prove that any configuration satisfying the conditions found in Problems 6 and 7 is solvable. One solution path is to force the parity of the edge and corner permutations to both be even, and then figure out how to do 3-cycles of edges and 3-cycles of corners separately.*

If you manage to solve Problem 8, you will have mathematically deduced how to solve a Rubik's cube! Bravo.

Problem 9. *In how many ways can we put together a disassembled Rubik's cube, assuming that you don't take apart the core that holds the center pieces together? What proportion of these are solvable?*

Problem 10 (NYCTC Fall 2023 Shortlist)**.** *You are given a solved Rubik's cube. Holding the cube in standard position (white center on top, green center in front), you are allowed to turn only the top, right, bottom, and left faces (no F or B moves). What proportion of the solvable cube states can you reach with this restricted move set?*