One Cycle to Rule Them All

ICIAM 2023

Giovanni Viglietta (University of Aizu)


CLASSIFICATION OF FINITE SIMPLE GROUPS
お前はもう死んでいる
PERMUTATION PUZZLE
なにぃっ！？

Topics

- Theory of permutation groups
- Applications to cycle-shift puzzles
- Applications to tilt-type puzzles

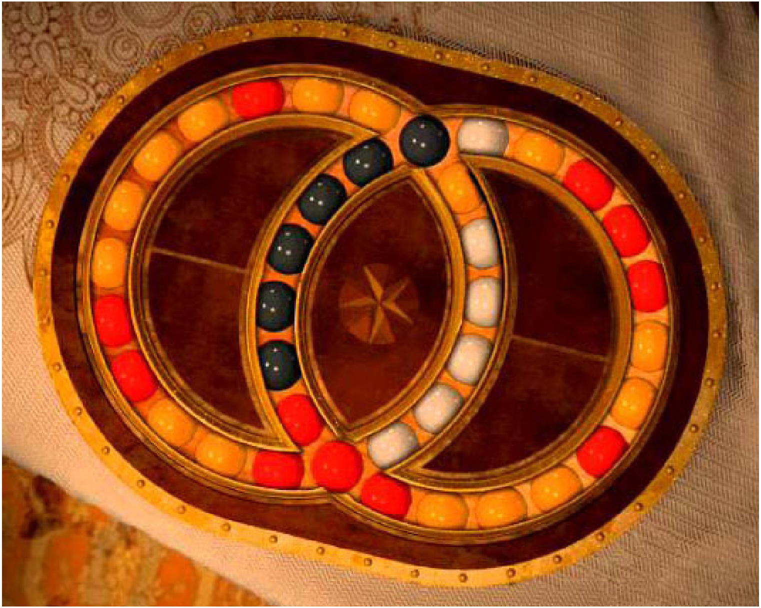
Permutation puzzles

Permutation puzzles

Permutation puzzles

Permutation puzzles

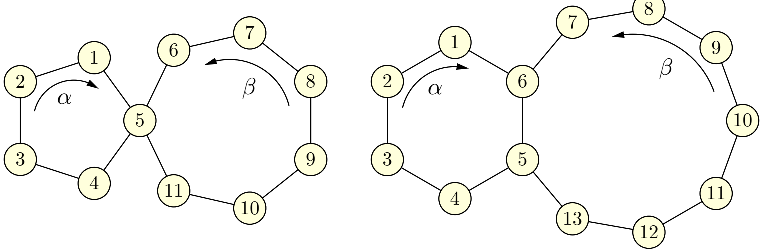
Permutation puzzles



Permutation puzzles

Cycle-shift puzzles

Consider these *cycle-shift puzzles*:



1-connected 2-connected

Our questions are:

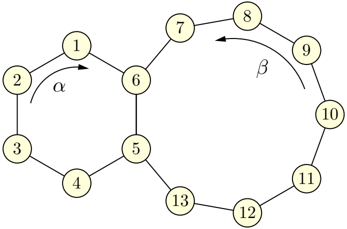
- What configurations are reachable from a given initial configuration? (I.e., what is the *configuration space*?)
- How can we get from an initial configuration to a goal configuration in a small number of moves?

Note: we assume that all tokens have <u>distinct colors</u> (or labels).

Generators

We have two permutations α and β, the **generators**:

- $\alpha = (1\ 2\ 3\ 4\ 5\ 6)$
- $\beta = (5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13)$



The set of permutations obtained by composing the generators in all possible ways is $\langle \alpha, \beta \rangle$, the group *generated* by α and β.

$$\langle \alpha, \beta \rangle = \{\, e,\ \alpha,\ \beta,\ \alpha\alpha,\ \alpha\beta,\ \beta\alpha,\ \beta\beta,\ \ldots,\ \beta\alpha\alpha\beta\beta\alpha\beta\beta\beta,\ \ldots \}$$

Finite groups of permutations

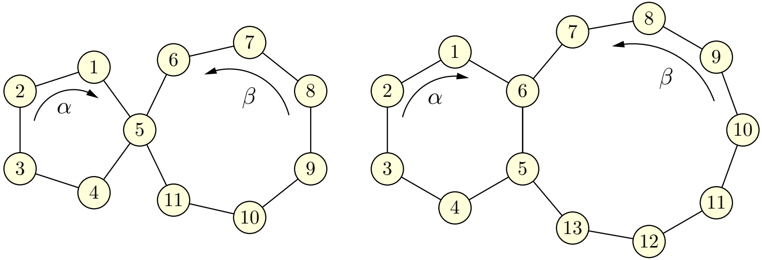
- A **finite permutation group** of *degree* n is a non-empty set of permutations of n objects that is closed under composition.
- The group of all permutations of n objects is called the **symmetric group** S_n.
- A permutation is **even** (resp. **odd**) if it is the result of an even (resp. odd) number of transpositions of pairs of objects.
- The group of all even permutations of n objects is called the **alternating group** A_n.
- In a finite permutation group, either all permutations are even or there is an equal number of even and odd permutations.

Parity of cycles

Observation

A cycle of length k is the composition of $k-1$ transpositions.

Example: $(1\ 2\ 3\ 4\ 5) = (1\ 2)(2\ 3)(3\ 4)(4\ 5)$.



So, the two cycles α and β generate a subgroup of A_n if and only if they <u>both</u> have odd length.

Can we prove that α and β generate *exactly* A_n or S_n?

Some known facts

Let a set of of generators K be given as input, and let $G = \langle K \rangle$.
These problems are solvable in polynomial time (Sims, 1970):

- Compute the *order* (i.e., size) of G.
- Decide if a given permutation π is in G.
- If $\pi \in G$, find an expression for π in terms of the generators.

On the other hand, the minimization problem is hard:

- If $\pi \in G$, finding the shortest sequence of generators whose composition is π is PSPACE-complete (Jerrum, 1985).
- If all the generators in K are cycles, the problem is NP-hard (Sai–V.–Uehara, 2022). Its PSPACE-completeness is open.

Some known facts

Under some conditions that are satisfied by cycle-shift puzzles,

- The length of a shortest generator sequence for π is upper bounded by a quasi-polynomial function of n (Helfgott–Seress, 2013).
- It is not known if there is a polynomial upper bound. If so, finding the shortest sequence of generators would be in NP.

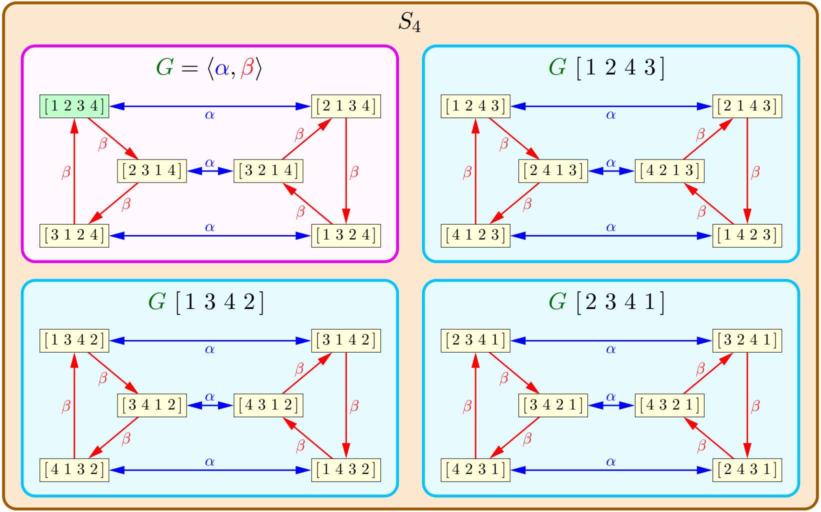
Most groups generated by two generators are S_n or A_n:

- Two random permutations of n objects generate either S_n or A_n with probability $1 - 1/n + O(n^{-2})$ (Babai, 1989).*
- The permutations π such that $\langle\, (1\ 2\ \ldots\ n),\, \pi \,\rangle$ is S_n or A_n have been characterized (Heath et al., 2009).

*This says nothing about the special case where the generators are cycles.

Coset structure

This is the **Cayley graph** of the subgroup $G \leq S_4$ generated by $\alpha = (1\ 2)$ and $\beta = (1\ 2\ 3)$, as well as the **cosets** of G.



Simple groups

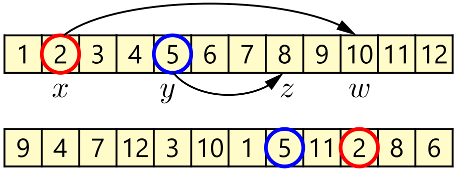
- Let G have a subgroup N. If the cosets of N in G form themselves a group, then N is a **normal** subgroup of G.
- In this case, the group formed by the cosets of N is called the **quotient group** G/N.
- If G has a normal subgroup N, one can learn everything about the structure of G by studying N and G/N.
- So, G can be broken down into **simple groups**, which are groups that have no (non-trivial) normal subgroups.
- Galois was the first to recognize the importance of simple groups and to prove that A_n **is simple** for all $n \neq 4$.
- Around 2004, the **classification of finite simple groups** has been completed.

Classification of finite simple groups

Application to permutation groups

Definition

A permutation group G on $\{1, 2, \ldots, n\}$ is **2-transitive** if, for every $1 \leq x, y, w, z \leq n$ with $x \neq y$ and $w \neq z$, there is a permutation $\pi \in G$ such that $\pi(x) = w$ and $\pi(y) = z$.



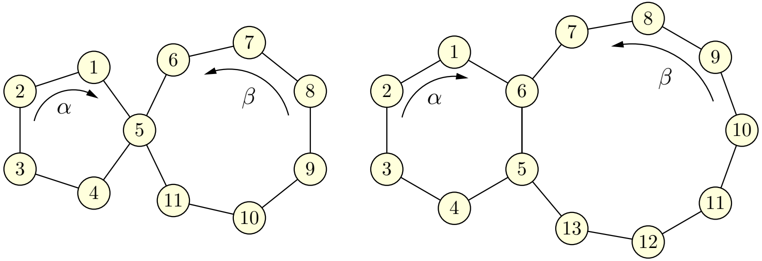
Theorem (Jones, 2014)

If a 2-transitive permutation group G on n objects contains a cycle[a] of length $n - 3$ or less, then G is either S_n or A_n.

[a] "One cycle to rule them all!"

Application to permutation groups

It is easy to prove that the permutation groups of 1-connected and 2-connected cycle-shift puzzles are **2-transitive**.



So, if there are $n > 6$ tokens in total, α and β generate either S_n or A_n, depending on whether one of them has even length.

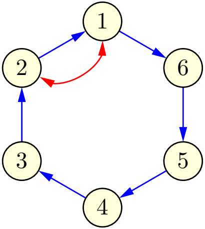
It still makes sense to look for constructive, efficient solutions.

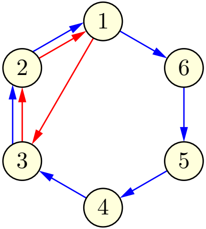
Generators of S_n and A_n

Lemma (Bubble Sort)

- $\langle (1\ 2\ \ldots\ n), (1\ 2) \rangle = S_n$.
- $\langle (1\ 2\ \ldots\ n), (1\ 2\ 3) \rangle \geq A_n$.

Any permutation in the group can be generated in $O(n^2)$ steps.





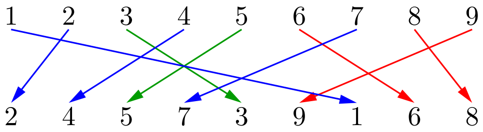
Therefore, if our α and β generate the cycles above, we can conclude that they generate all of S_n or A_n.

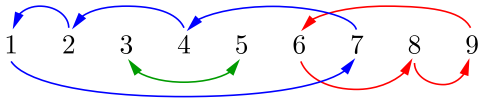
Cycle decomposition

Observation

Every permutation can be expressed as the composition of disjoint cycles in a unique way (up to reordering).

Example:





So, the permutation can be written as $(1\ 2\ 4\ 7)(3\ 5)(9\ 8\ 6)$.

Conjugation

Definition

The permutation π, **conjugated** by σ, is the permutation $\sigma\pi\sigma^{-1}$.

The same operation is done in linear algebra when changing coordinates: a linear transformation defined by a matrix A can also be expressed as PAP^{-1}, where P is a nonsingular matrix defining a *change of basis*.

Lemma

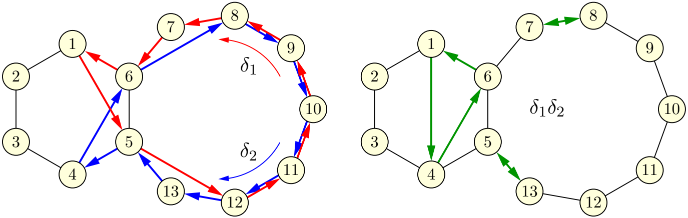
Conjugation preserves the **cycle structure** *of permutations.*

Example: $(3\ 5\ 7)\,(1\ 3\ 4)(2\ 6)(5\ 7)\,(7\ 5\ 3) = (1\ 5\ 4)(2\ 6)(7\ 3)$.

$\Longrightarrow$ Conjugation allows us to "move cycles around" in a puzzle...

Application of conjugation

In a 2-connected puzzle, conjugating β by $\beta^{-1}\alpha$ and β^{-1} by $\beta\alpha^{-1}$, we obtain two cycles δ_1 and δ_2 of the same length, going in opposite directions:

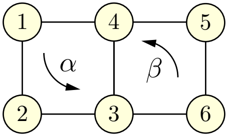


Their composition $\delta_1\delta_2$ is a 3-cycle plus two transpositions.
So, $(\delta_1\delta_2)^2$ is a 3-cycle.

Once we have a 3-cycle, we can move it around and do Bubble Sort (again, by conjugation).

Recap on cycle-shift puzzles

Jones' theorem does not apply to this small puzzle, which is a genuine exception and requires an ad-hoc analysis:



Theorem (*Sai–V.–Uehara, IEICE Trans Inf Syst, 2022*)

In a 1-connected or 2-connected puzzle, α and β generate:

- A_n *if both α and β have odd length;*
- S_n *if α or β has even length, with one exception:*
- *if the puzzle is 2-connected and α and β have length 4, they generate a group isomorphic to S_5 (as opposed to S_6).*

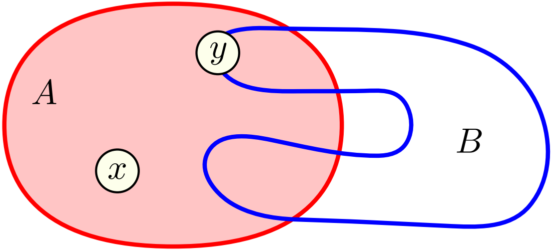
Any permutation in the group can be generated in $O(n^2)$ steps.

More than two generators

What if the cycle-shift puzzle has more than two generators?

Lemma

If G acts 2-transitively on A and contains a cycle spanning B, with $A \cap B \neq \emptyset$ and $A \setminus B \neq \emptyset$, then G acts 2-transitively on $A \cup B$.



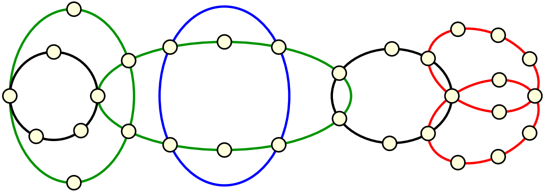
Starting from a 1-connected or 2-connected puzzle, we can attach any number of additional cycles, and Jones' theorem still applies.

Generalized cycle-shift puzzles

Let $\mathcal{C}$ be a set of cycles, and let $\hat{G} = (\mathcal{C}, \mathcal{E})$ be the graph where $\{C_1, C_2\} \in \mathcal{E}$ if C_1 and C_2 induce a 1- or a 2-connected puzzle.

$\mathcal{C}$ forms a *proper cycle-shift puzzle* if there is $\mathcal{C}' \subseteq \mathcal{C}$ such that:

- $\mathcal{C}'$ contains at least two cycles.
- The induced subgraph $\hat{G}[\mathcal{C}']$ is connected.
- Each token is contained in at least one cycle in $\mathcal{C}'$.

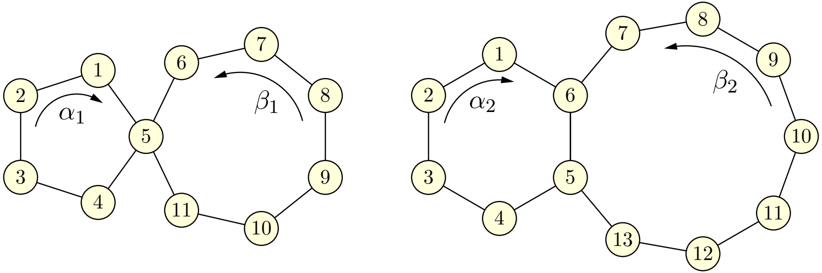


Theorem (*Sai–V.–Uehara, IEICE Trans Inf Syst, 2022*)

The configuration group of a proper cycle-shift puzzle with more than 6 tokens is A_n if all cycles have odd length, and S_n otherwise. Any permutation in the group can be generated in $O(n^5)$ steps.

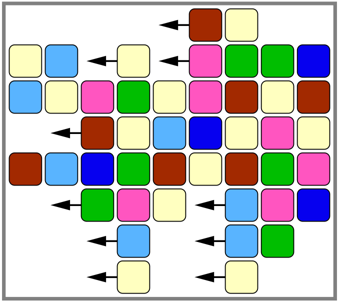
Open problems

- What pairs of cycles generate a 2-transitive group?
- In the puzzle below, what if α_1 and α_2 are constrained to shift in unison (*entangled*), and so are β_1 and β_2?



- In general, if we know the permutation groups of two cycle-shift puzzles (with the same number of cycles), what can we say about the group of their *entangled sum*?

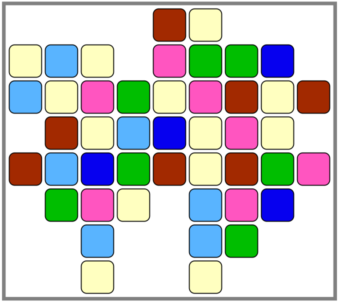
Tilt-1 puzzle: There are some colored tokens on a board. A move consists in "tilting" the board in one direction, letting all tokens slide by one position.



Problem: Starting from a given initial configuration, what other configurations can be reached?

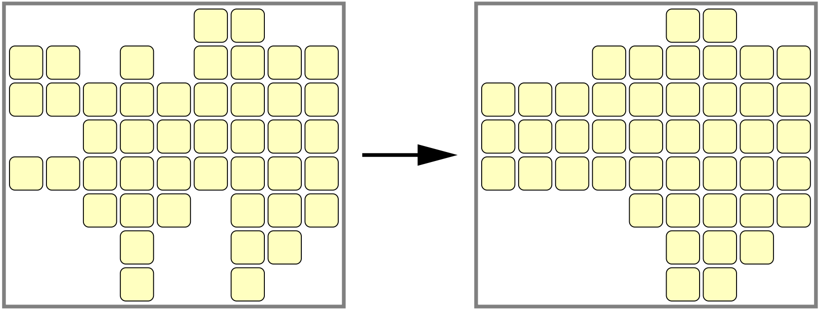
Tilt-1 puzzle

Tilt-1 puzzle: There are some colored tokens on a board. A move consists in "tilting" the board in one direction, letting all tokens slide by one position.



Problem: Starting from a given initial configuration, what other configurations can be reached?

Compact configurations

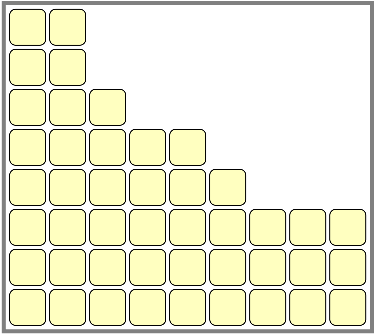


After some moves, the configuration tends to become orthogonally convex, where each row (resp. column) of length k is contained in the projection of every row (resp. column) of length $\geq k$.
We call such a configuration *compact*.

Canonical configurations

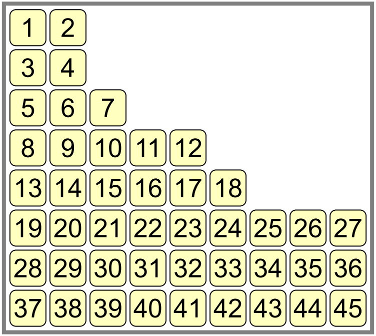
Lemma

Any move performed from a compact configuration is reversible.



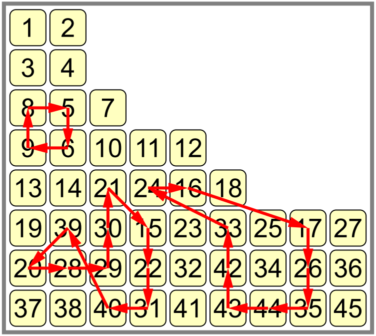
Thus, without loss of generality, we will assume the configuration to be a compact "staircase" with a full leftmost column and a full bottommost column. We call such a configuration *canonical*.

Canonical configurations



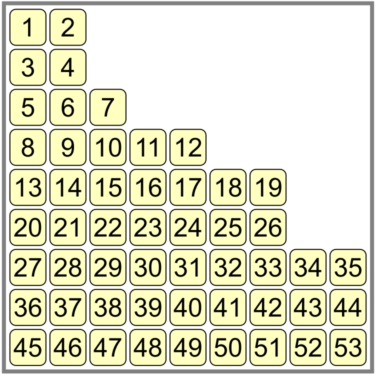
Let us assign unique labels to the tokens. After playing some moves and restoring a canonical configuration, we obtain a permutation of these tokens.

Canonical configurations



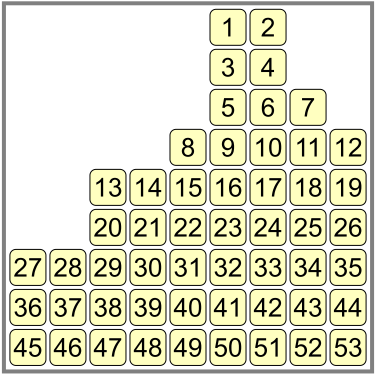
The problem is to study the permutations that can be achieved.

Generating cycles



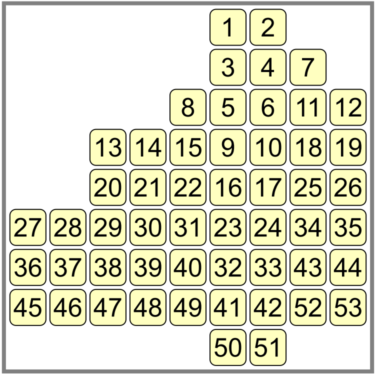
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



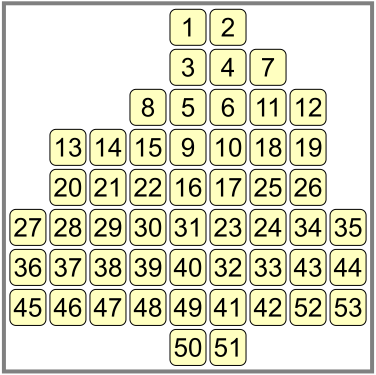
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



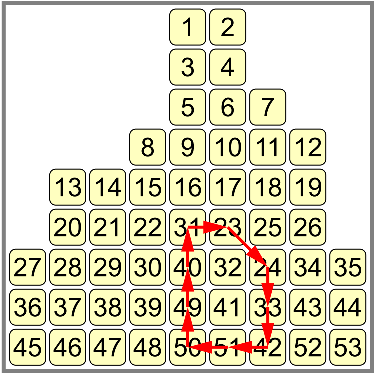
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



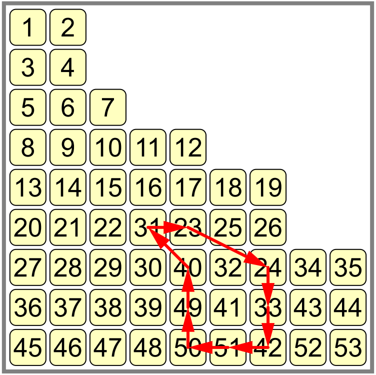
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



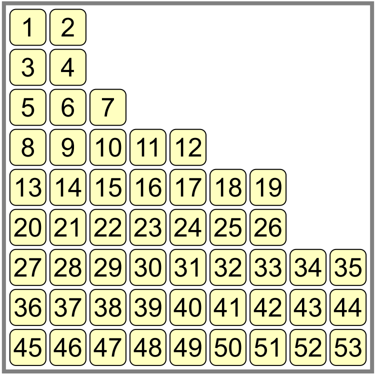
The sequence of moves $\mathrm{R}^k\mathrm{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Generating cycles



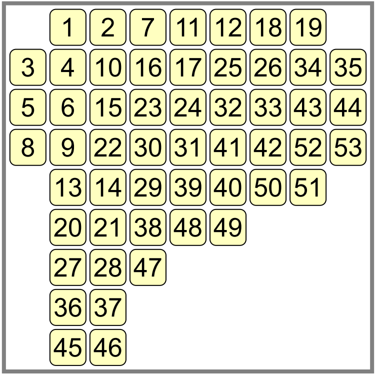
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles

1	2	7	11	12	18	19		
3	4	10	16	17	25	26	34	35
5	6	15	23	24	32	33	43	44
8	9	22	30	31	41	42	52	53
13	14	29	39	40	50	51		
20	21	38	48	49				
27	28	47						
36	37							
45	46							

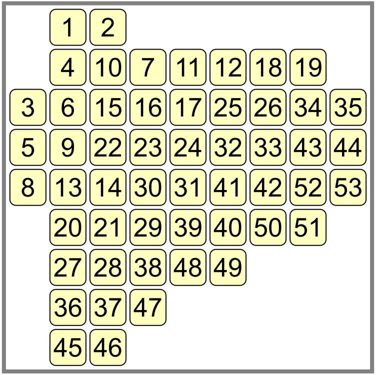
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles



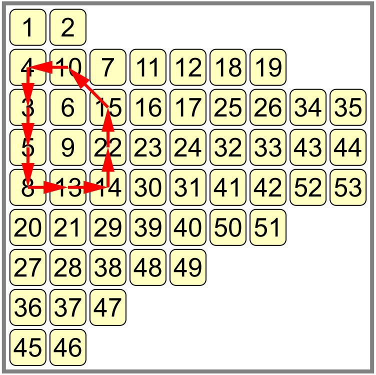
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles



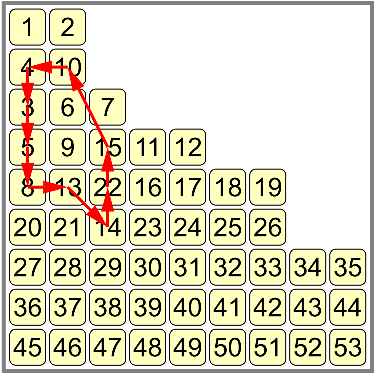
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles



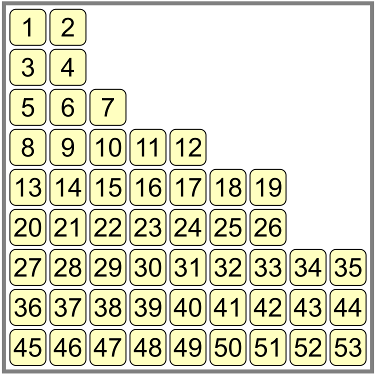
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles



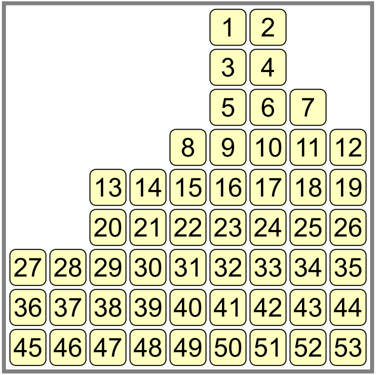
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Generating cycles



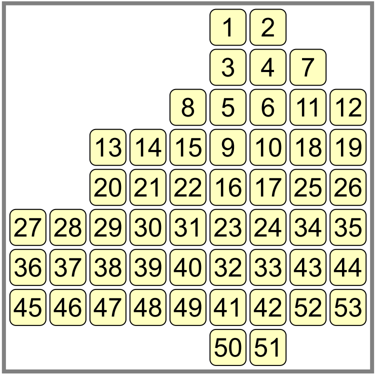
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



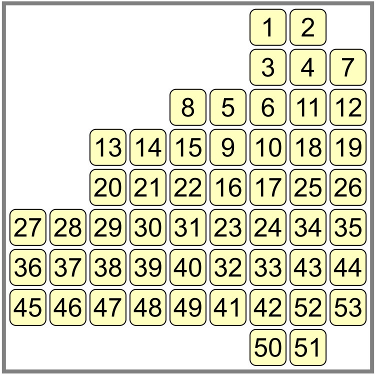
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



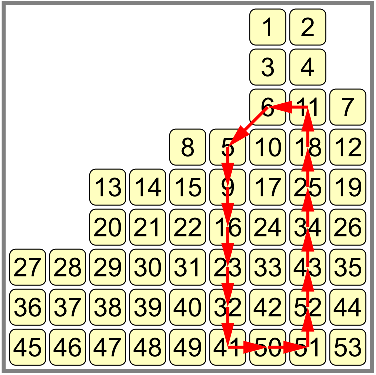
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



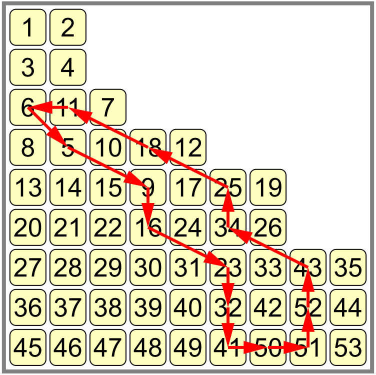
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



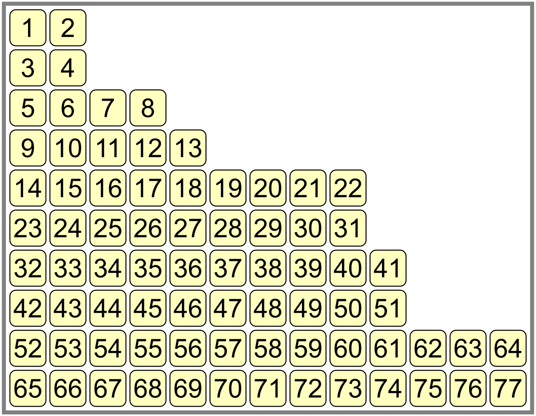
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



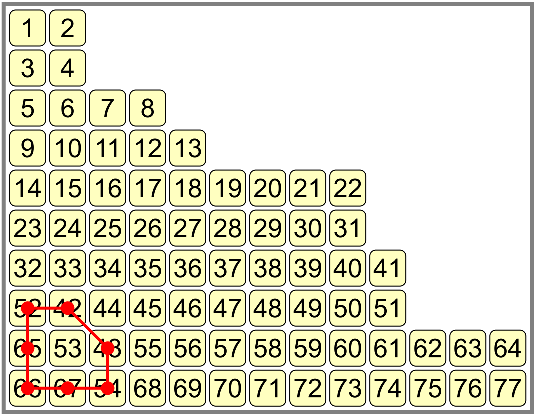
The sequence of moves R^{k}URDLL^k, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Generating cycles



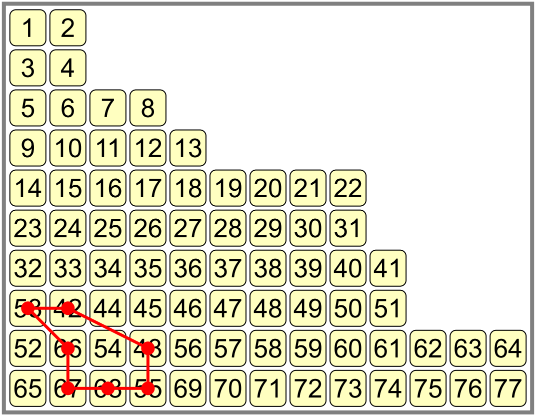
These cycles collectively span all tokens.

Generating cycles

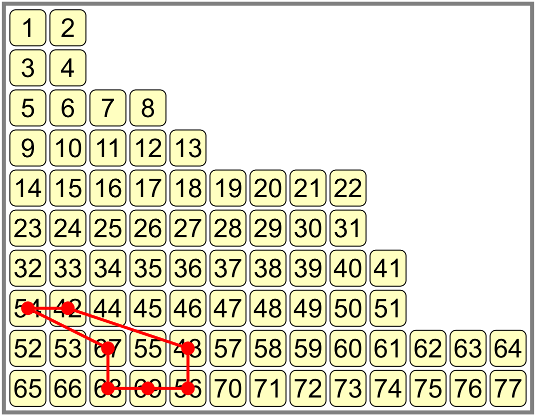


These cycles collectively span all tokens.

Generating cycles

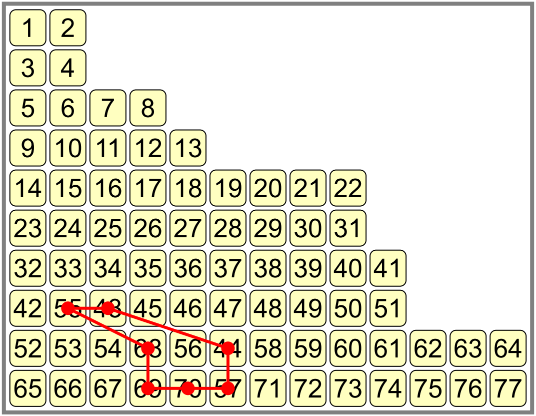


These cycles collectively span all tokens.



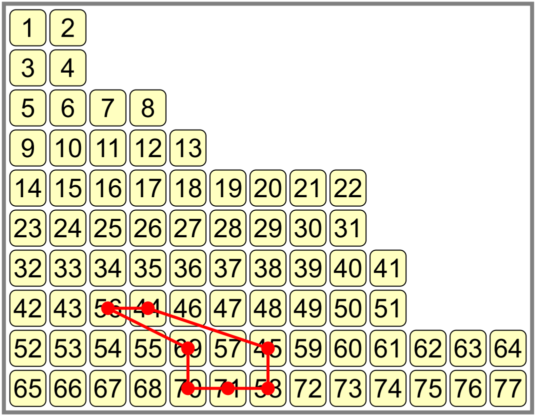
These cycles collectively span all tokens.

Generating cycles



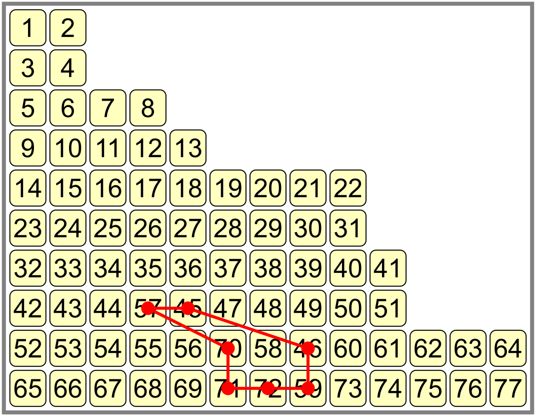
These cycles collectively span all tokens.

Generating cycles



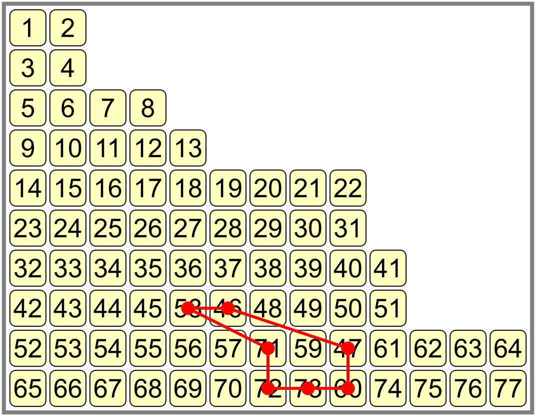
These cycles collectively span all tokens.

Generating cycles



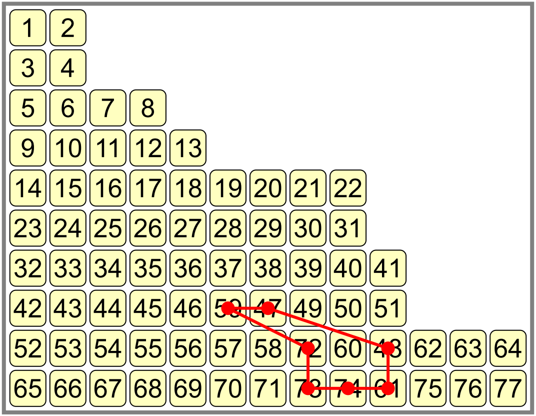
These cycles collectively span all tokens.

Generating cycles



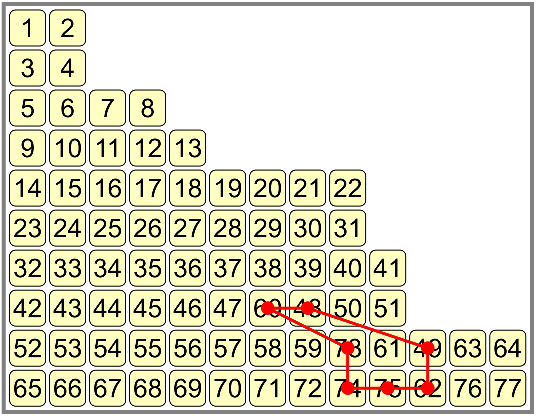
These cycles collectively span all tokens.

Generating cycles



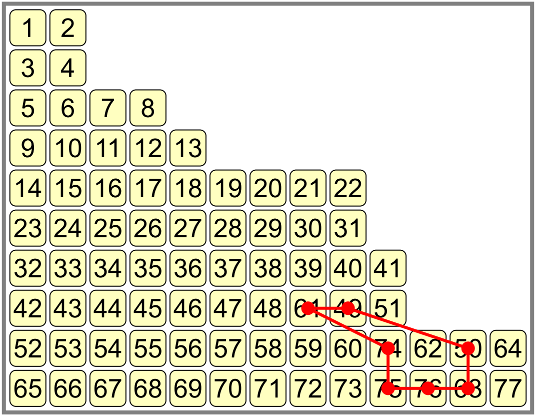
These cycles collectively span all tokens.

Generating cycles



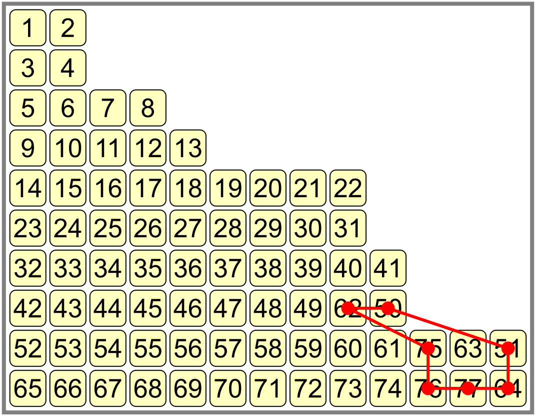
These cycles collectively span all tokens.

Generating cycles



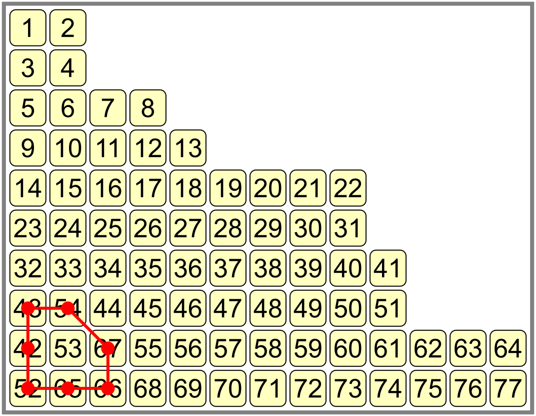
These cycles collectively span all tokens.

Generating cycles



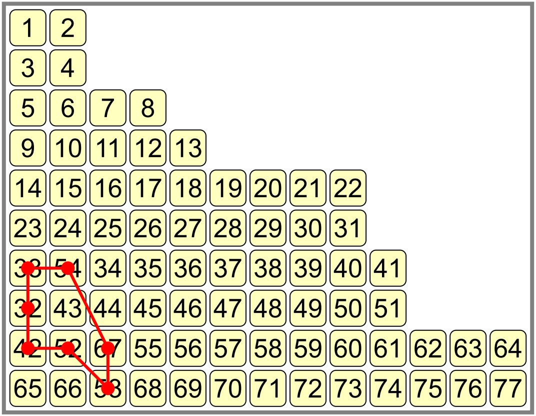
These cycles collectively span all tokens.

Generating cycles



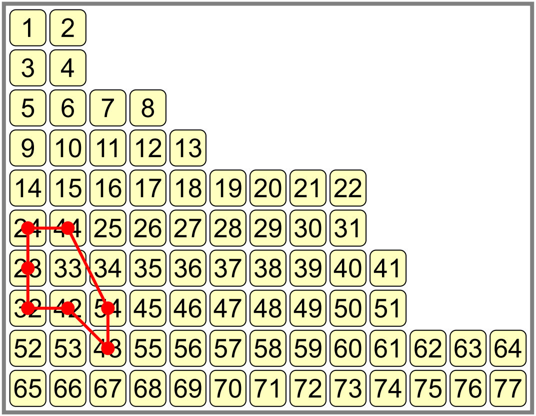
These cycles collectively span all tokens.

Generating cycles



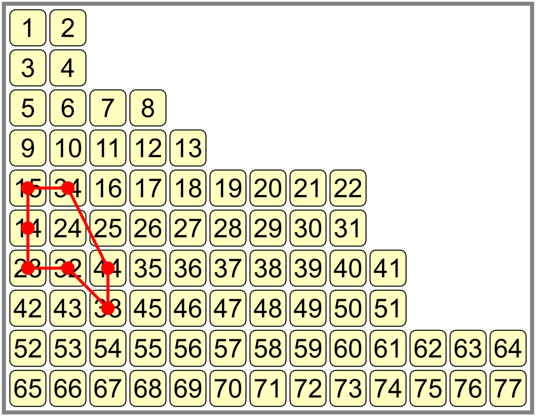
These cycles collectively span all tokens.

Generating cycles



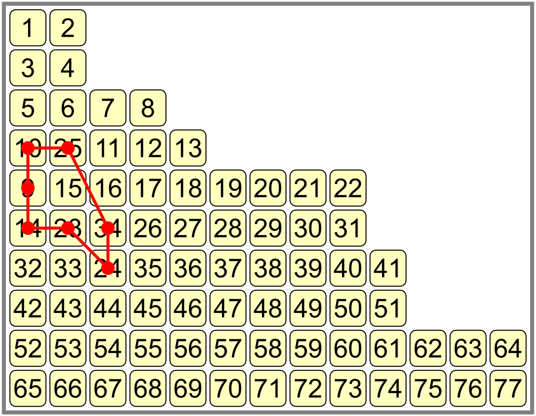
These cycles collectively span all tokens.

Generating cycles



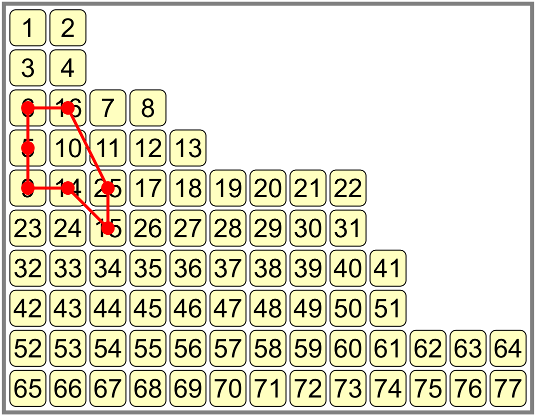
These cycles collectively span all tokens.

Generating cycles



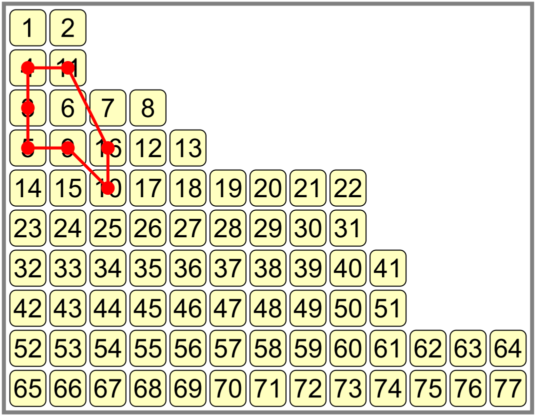
These cycles collectively span all tokens.

Generating cycles



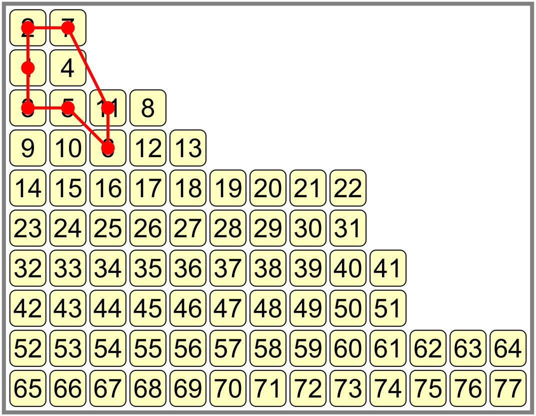
These cycles collectively span all tokens.

Generating cycles



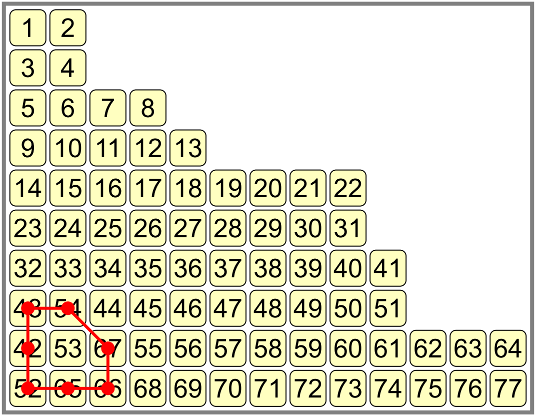
These cycles collectively span all tokens.

Generating cycles

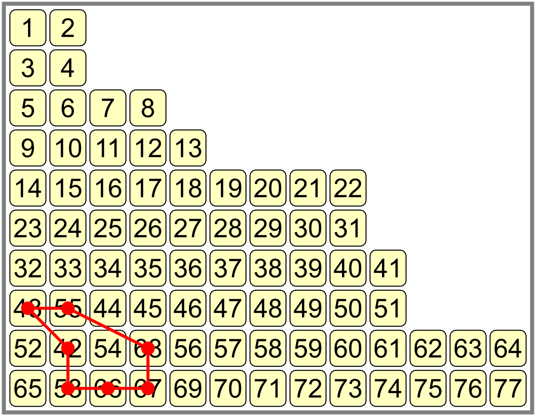


These cycles collectively span all tokens.

Generating cycles

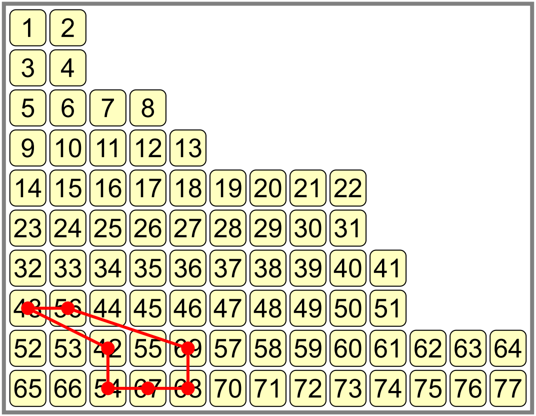


These cycles collectively span all tokens.



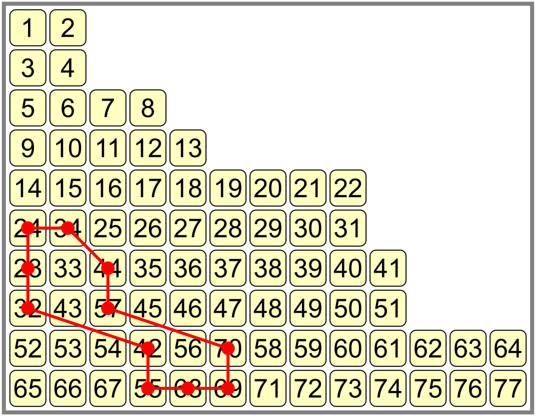
These cycles collectively span all tokens.

Generating cycles



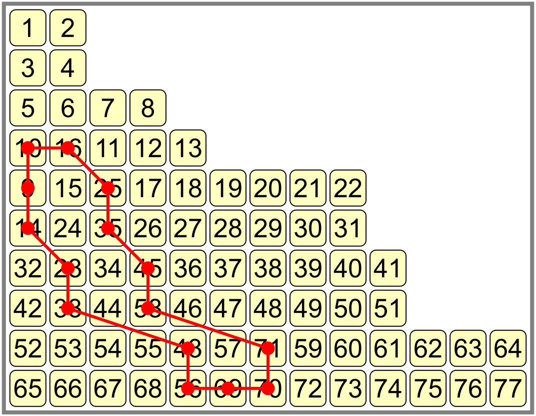
These cycles collectively span all tokens.

Generating cycles



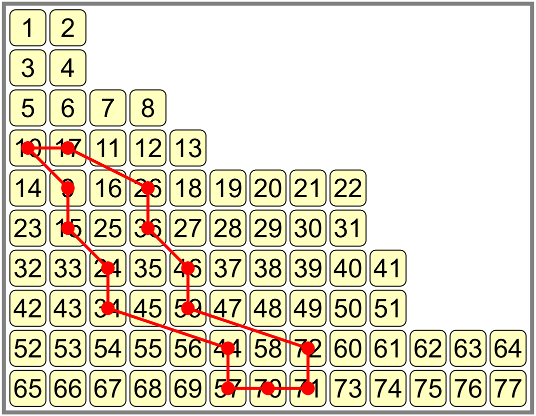
These cycles collectively span all tokens.

Generating cycles



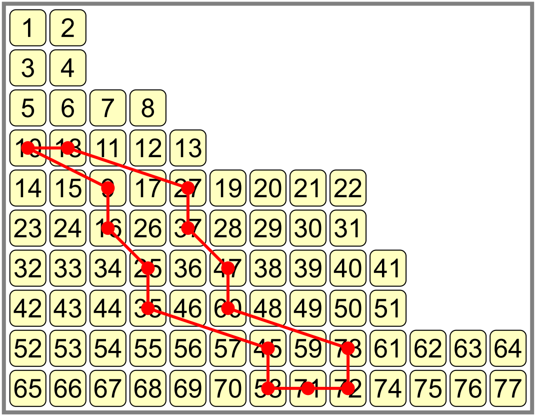
These cycles collectively span all tokens.

Generating cycles



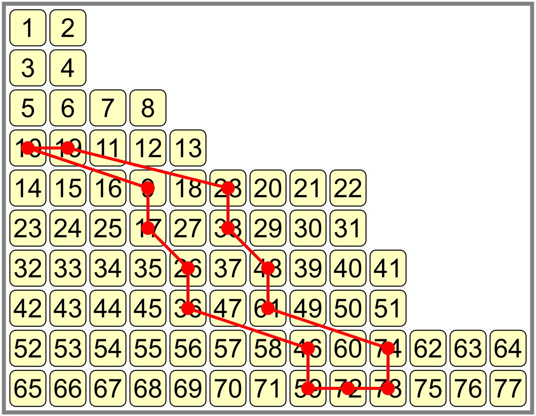
These cycles collectively span all tokens.

Generating cycles



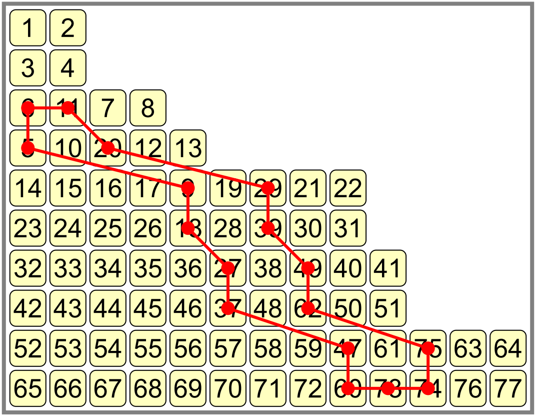
These cycles collectively span all tokens.

Generating cycles

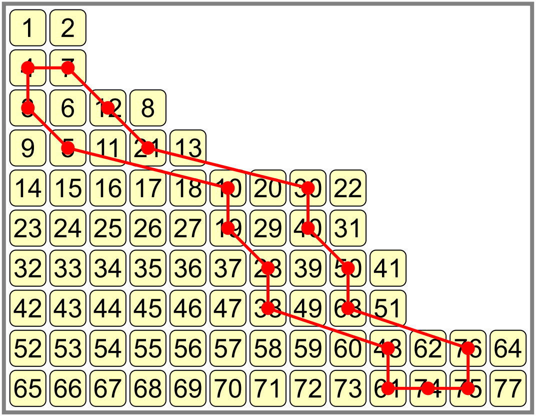


These cycles collectively span all tokens.

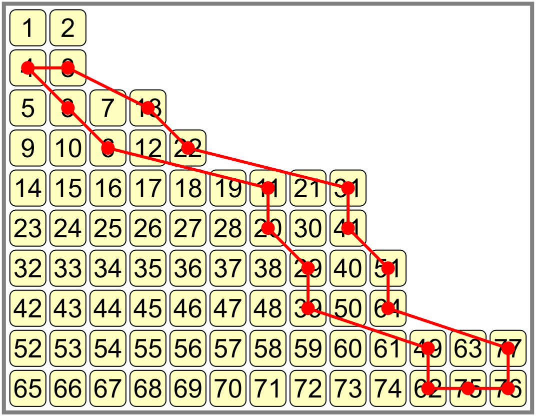
Generating cycles



These cycles collectively span all tokens.



These cycles collectively span all tokens.

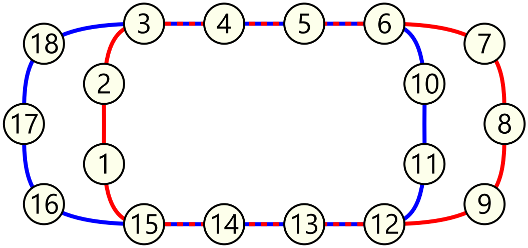


These cycles collectively span all tokens.

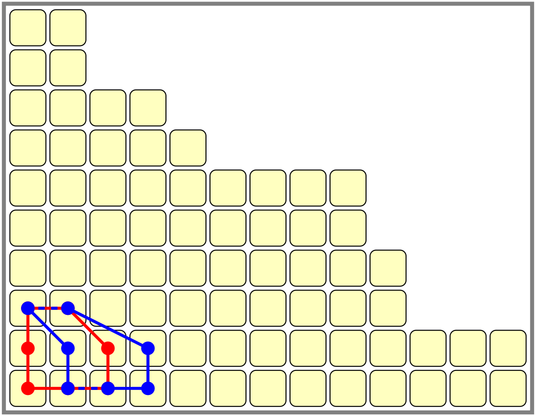
More 2-transitive groups

Lemma

If G contains two cycles interconnected as in the figure, then G acts 2-transitively on the items spanned by the two cycles.

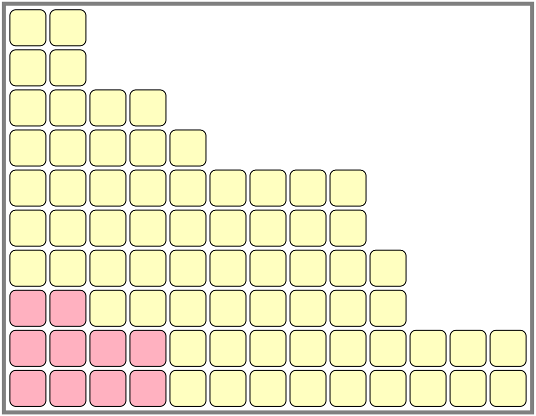


Generating all even permutations



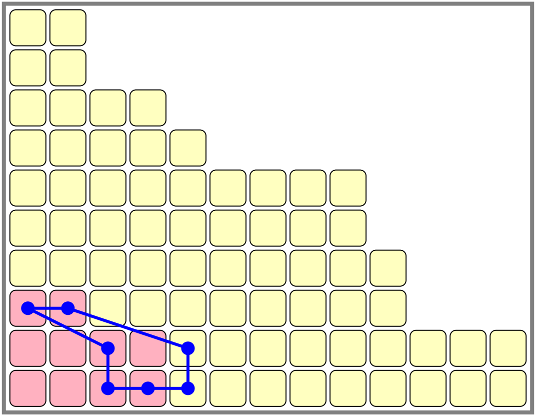
So, these two cycles determine a 2-transitive permutation group acting on the tokens they span.

Generating all even permutations



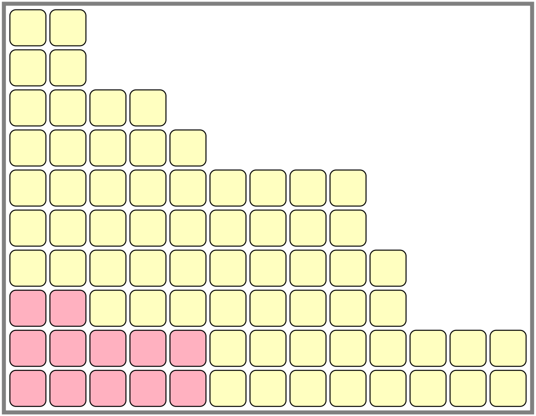
So, these two cycles determine a 2-transitive permutation group acting on the tokens they span.

Generating all even permutations



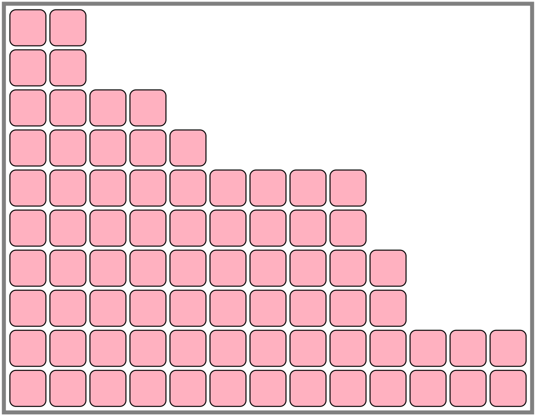
Then we can extend this set to the tokens spanned by the next cycle.

Generating all even permutations



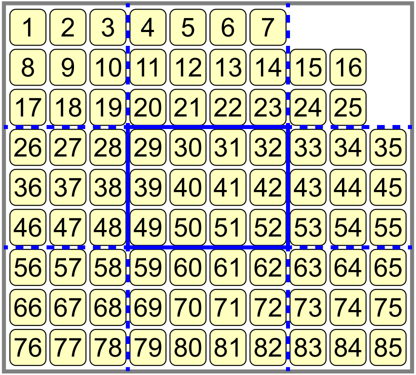
Proceeding in this fashion, we conclude that our cycles act 2-transitively on all the tokens they span.

Generating all even permutations



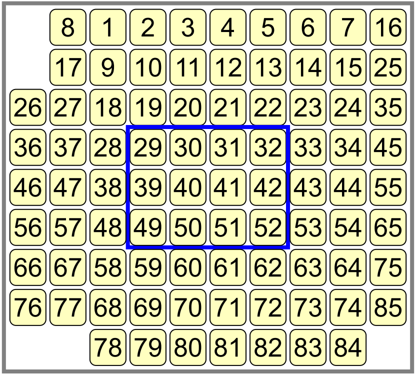
Thus, by Jones' theorem, we can generate all even permutations on the tokens spanned by our cycles.

Unmovable core

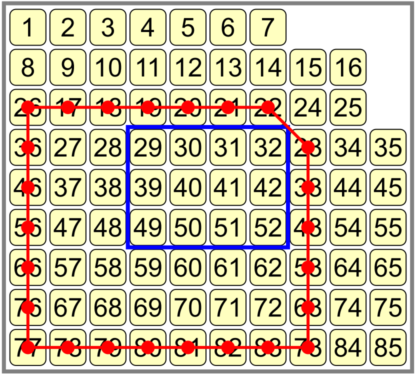


In general, if more than half of the rows and more than half of the columns are full, there is a *central core* that is impossible to move.

Unmovable core

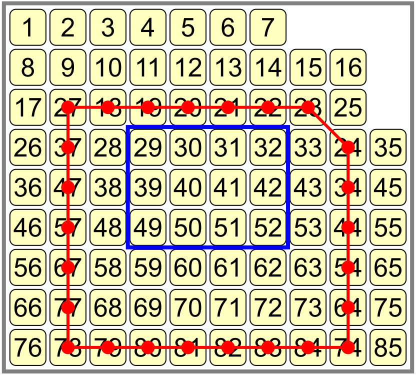


In general, if more than half of the rows and more than half of the columns are full, there is a *central core* that is impossible to move.

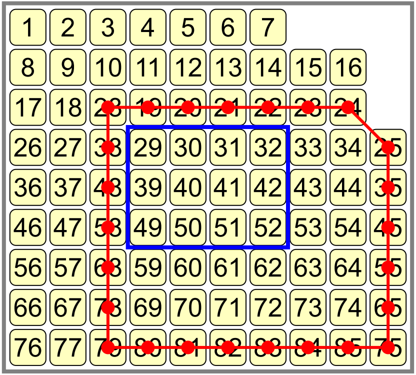


However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Unmovable core

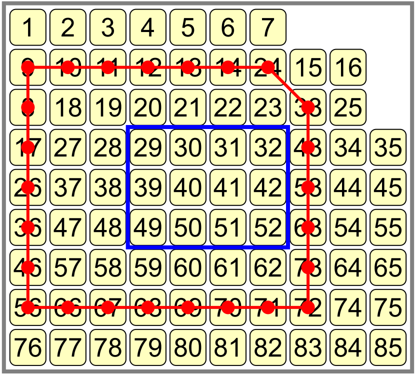


However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.



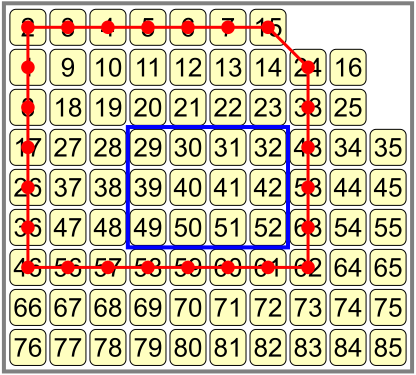
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Unmovable core

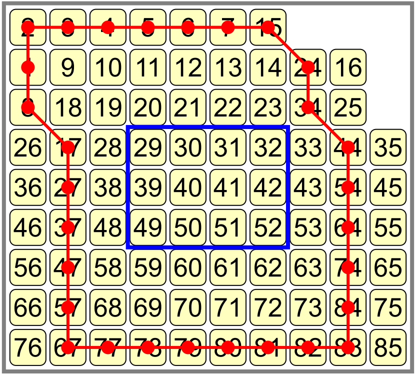


However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Unmovable core

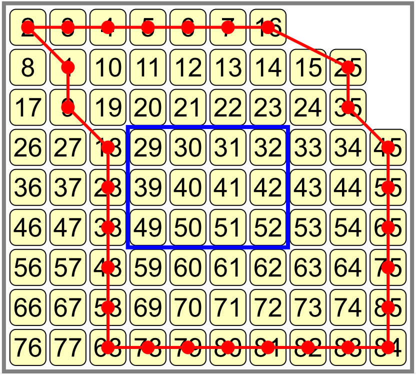


However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.



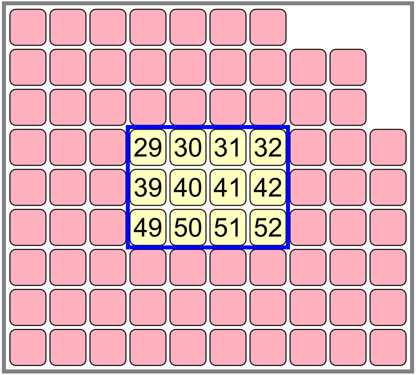
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Unmovable core



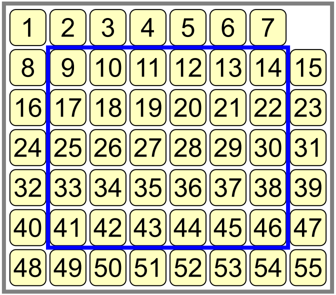
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Unmovable core



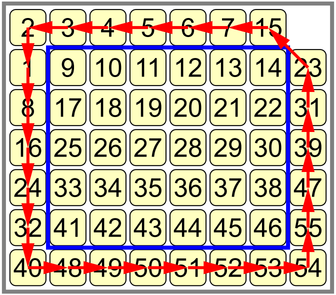
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Special cases



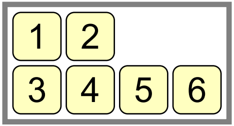
An exception is when there is exactly one empty cell. In this case, the permutation group is easily seen to be cyclic.

Special cases



An exception is when there is exactly one empty cell. In this case, the permutation group is easily seen to be cyclic.

Special cases

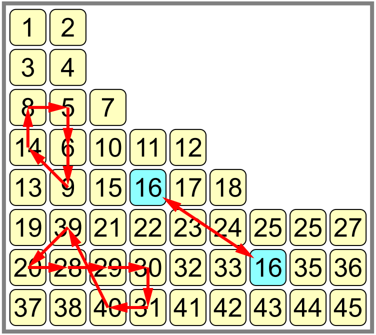


The only other exception is the puzzle with a 4×2 bounding box and two empty cells. In this case, the permutation group has order 60 and is isomorphic to the alternating group A_5 (not A_6!).*

*Trivia: S_6 is the only S_n having S_{n-1} and A_{n-1} as *transitive* subgroups.

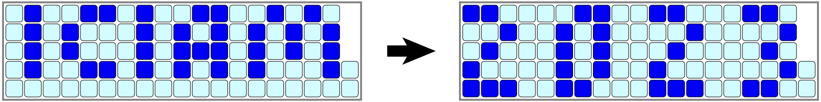
Non-unique labels

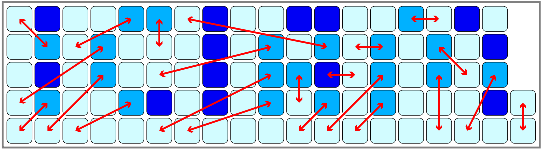
What if not all labels are unique? Note that we can transpose any two equal (non-core) labels "for free".



So, in order to generate any odd permutation π of the non-core labels, we can append a transposition $(x\ y)$ of any two equal non-core labels to reduce it to the even permutation $\pi \circ (x\ y)$.

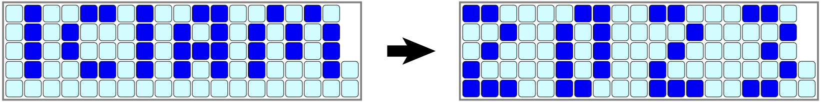
Non-unique labels

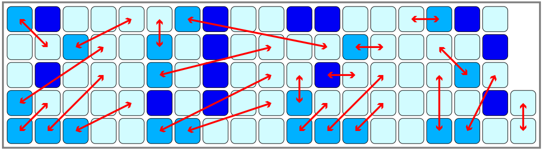




In this puzzle, since the core is empty and some tokens are equal, *all* permutations are possible (including the odd ones).

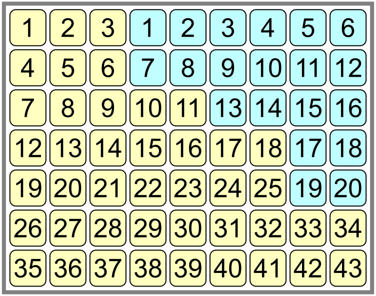
Non-unique labels





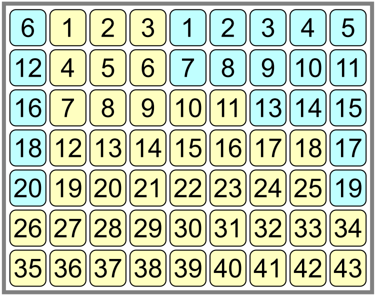
In this puzzle, since the core is empty and some tokens are equal, *all* permutations are possible (including the odd ones).

Permutations are even



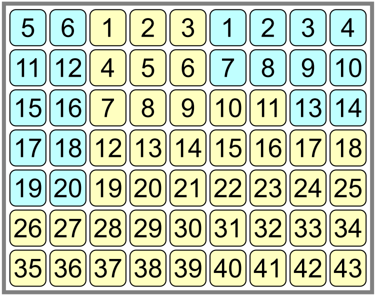
We still have to prove that any permutation obtainable in the Tilt-1 puzzle must be even. It will be convenient to label the empty cells, too.

Permutations are even



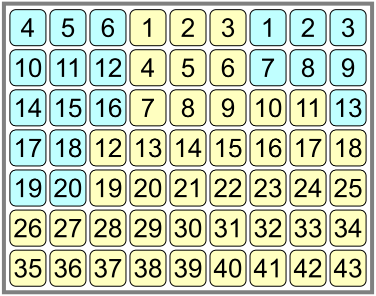
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutations are even



Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutations are even



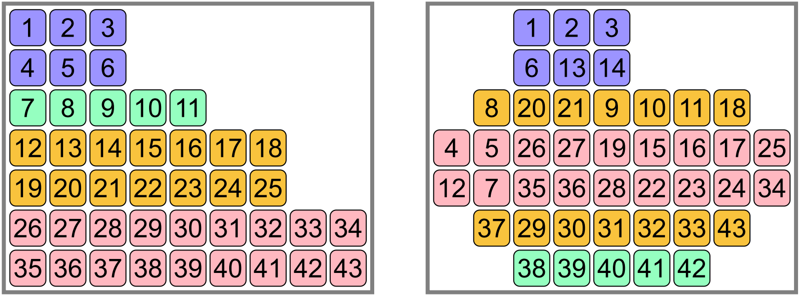
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutations are even

10	11	12	1	2	3	7	8	9
14	15	16	4	5	6	10	11	13
17	18	12	7	8	9	16	17	18
19	20	19	13	14	15	23	24	25
26	27	28	20	21	22	32	33	34
35	36	37	29	30	31	41	42	43
4	5	6	38	39	40	1	2	3

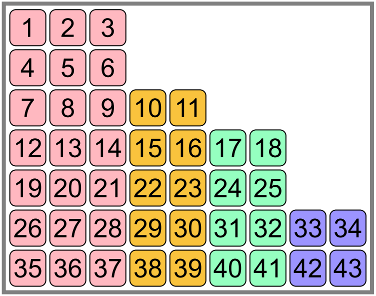
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

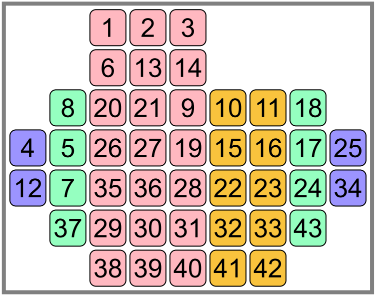
Permutations are even



Observe that all rows of the same length must always be aligned. Moreover, for every move that pushes all rows of some length to the right, there must be a move that pushes all of them to the left.

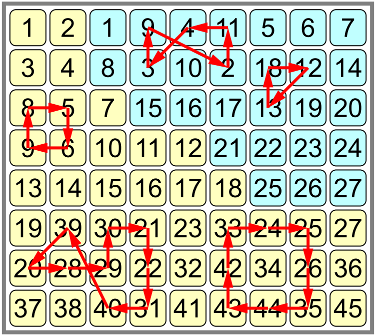
Permutations are even





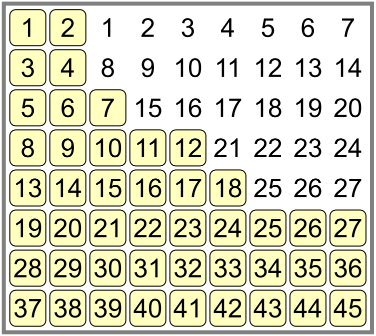
The same holds for columns and up/down moves. Thus, if we decompose every move into cycles (involving both tokens and empty cells), we see that every cycle must have a matching cycle of the same length.

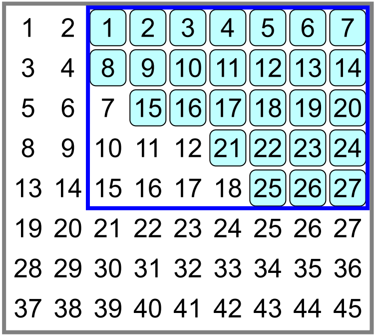
Permutations are even



So, if we restore a canonical configuration, the overall permutation must be <u>even</u>. We still need to prove that the same permutation, restricted to the tokens (equiv., to the empty cells), is also even.

Primal and dual puzzles





Let us isolate the empty cells, and treat them as a "dual puzzle".

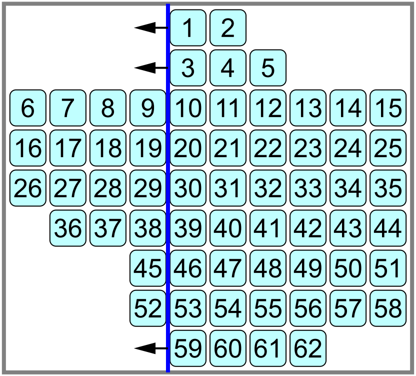
Primal and dual puzzles

16	21	22	23	24	**7**	**11**	8	15
25	26	27	**1**	**2**	**10**	**16**	**12**	**18**
8	**9**	**21**	**3**	**4**	**15**	**25**	**17**	**27**
13	**14**	**30**	**5**	**6**	**24**	**34**	**26**	**36**
19	**20**	**39**	**22**	**23**	**32**	**33**	**35**	**45**
4	5	6	7	**31**	**40**	**41**	**43**	**44**
3	11	12	13	14	**28**	**29**	**42**	2
10	17	18	19	20	**37**	**38**	1	9

40	41	43	44	**4**	**5**	**6**	**7**	31
28	29	42	**2**	**3**	**11**	**12**	**13**	**14**
37	38	**1**	**9**	**10**	**17**	**18**	**19**	**20**
7	11	**8**	**15**	**16**	**21**	**22**	**23**	**24**
10	16	12	18	**25**	**26**	**27**	1	2
15	25	17	27	8	9	21	3	4
24	34	26	36	13	14	30	5	6
32	33	35	45	19	20	39	22	23

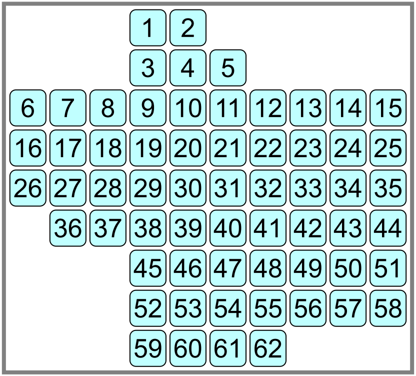
As we play the primal puzzle, we are also playing the dual puzzle.

Primal and dual puzzles



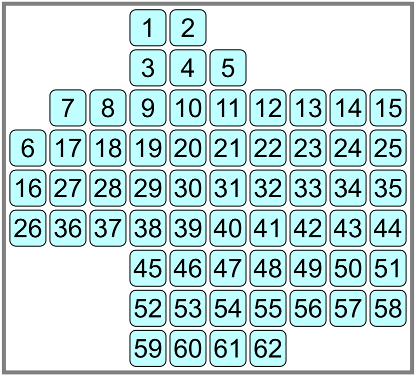
The rules of the dual puzzle are slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Primal and dual puzzles



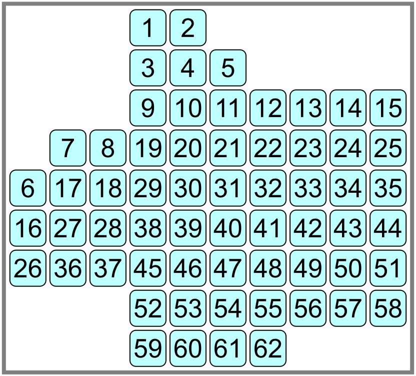
The rules of the dual puzzle are slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Primal and dual puzzles



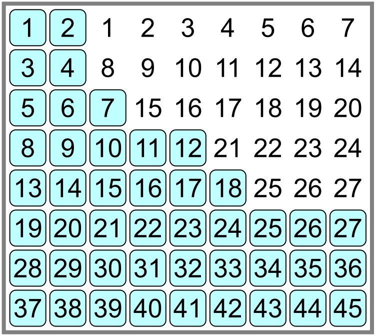
The rules of the dual puzzle are slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

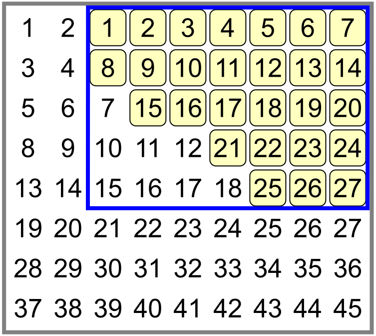
Primal and dual puzzles



The rules of the dual puzzle are slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Primal and dual puzzles





The dual puzzle of a dual-type puzzle is again a primal-type puzzle.

Primal and dual puzzles

7	1	2	1	2	3	4	5	6
14	3	4	8	9	10	11	12	13
5	6	7	15	16	17	18	19	20
8	9	10	11	12	21	22	23	24
13	14	15	16	17	18	25	26	27
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45

1	2	1	2	3	4	5	6	7
3	4	8	9	10	11	12	13	14
6	7	15	16	17	18	19	20	5
9	10	11	12	21	22	23	24	8
14	15	16	17	18	25	26	27	13
20	21	22	23	24	25	26	27	19
29	30	31	32	33	34	35	36	28
38	39	40	41	42	43	44	45	37

The dual puzzle of a dual-type puzzle is again a primal-type puzzle.

Primal and dual puzzles

7	1	2	1	2	3	11	12	13
14	3	4	8	9	10	18	19	20
5	6	7	15	16	17	22	23	24
8	9	10	11	12	21	25	26	27
13	14	15	16	17	18	25	26	27
19	20	21	22	23	24	34	35	36
28	29	30	31	32	33	43	44	45
37	38	39	40	41	42	4	5	6

38	39	40	41	42	4	5	6	37
1	2	1	2	3	11	12	13	7
3	4	8	9	10	18	19	20	14
6	7	15	16	17	22	23	24	5
9	10	11	12	21	25	26	27	8
14	15	16	17	18	25	26	27	13
20	21	22	23	24	34	35	36	19
29	30	31	32	33	43	44	45	28

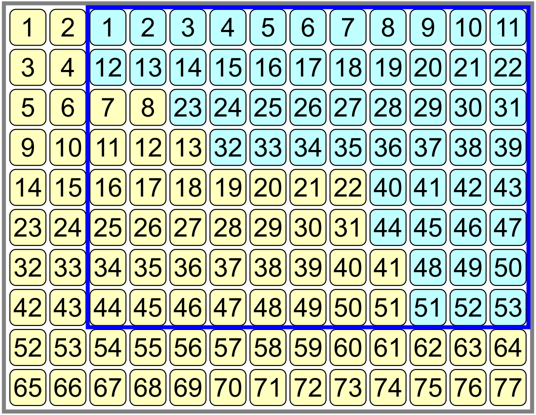
The dual puzzle of a dual-type puzzle is again a primal-type puzzle.

Permutations are even

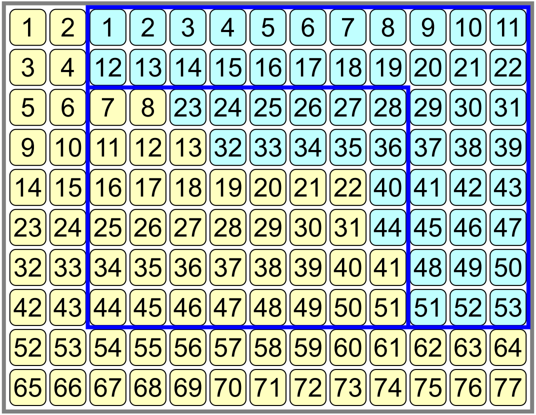
1	2	1	2	3	4	5	6	7	8	9	10	11
3	4	12	13	14	15	16	17	18	19	20	21	22
5	6	7	8	23	24	25	26	27	28	29	30	31
9	10	11	12	13	32	33	34	35	36	37	38	39
14	15	16	17	18	19	20	21	22	40	41	42	43
23	24	25	26	27	28	29	30	31	44	45	46	47
32	33	34	35	36	37	38	39	40	41	48	49	50
42	43	44	45	46	47	48	49	50	51	51	52	53
52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77

Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutations are even

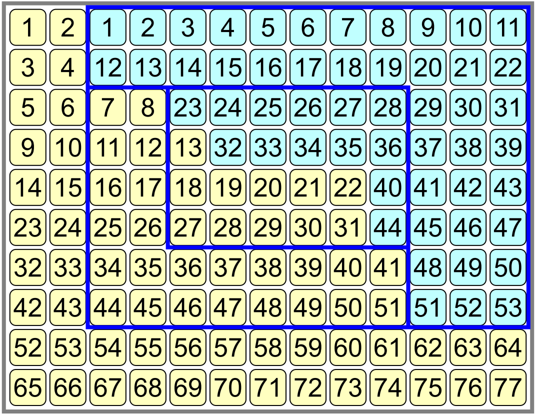


Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.



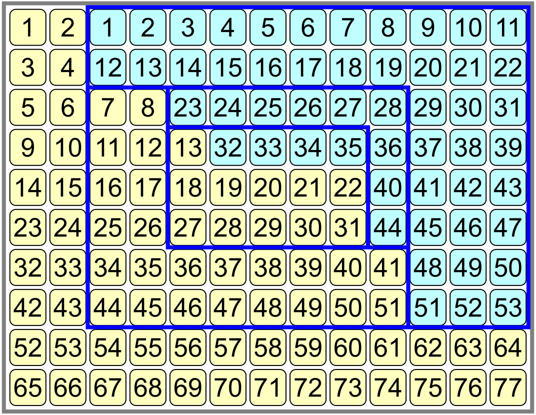
Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutations are even



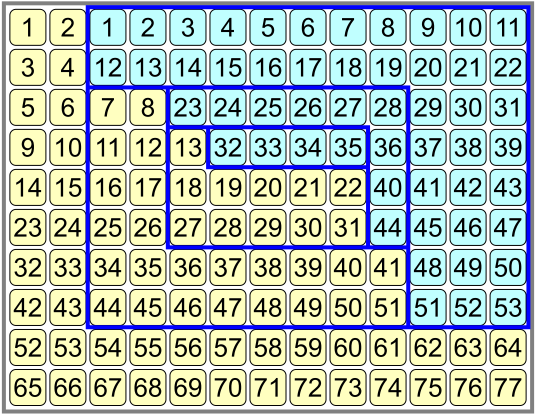
Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutations are even



Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutations are even



Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Conclusion and open problems

Theorem (*Akitaya–Löffler–V., FUN 2022*)

If the configuration is compact, then the possible permutations are:

- *If exactly one cell is empty, the cycles of the non-core tokens.*
- *If there are 6 tokens and 2 empty cells, a group isomorphic to A_5.*
- *Otherwise, if all non-core tokens have distinct labels, all the even permutations of the non-core tokens' labels.*
- *Otherwise, all permutations of the non-core tokens' labels.*

- Can we generalize this concept of duality to other puzzles?
- Consider the Tilt-∞ puzzle, where each move slides the tiles maximally. In this puzzle, in the cycle decomposition of a permutation, the even-length cycles seem to appear in pairs of equal length. Is this always true?