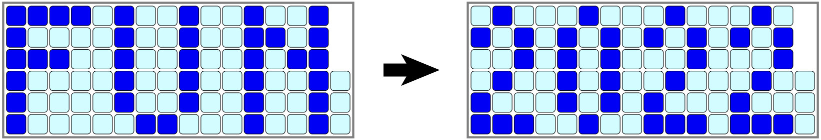
Pushing Blocks by Sweeping Lines

FUN 2022

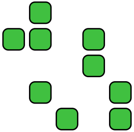
Hugo Akitaya, Maarten Löffler, and Giovanni Viglietta

Favignana – June 1, 2022



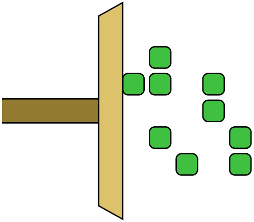
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



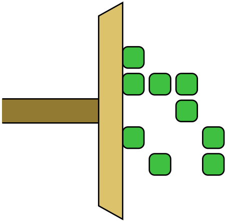
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



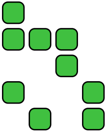
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



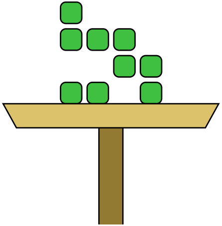
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



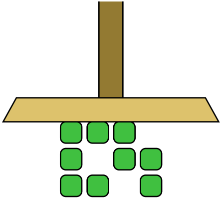
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



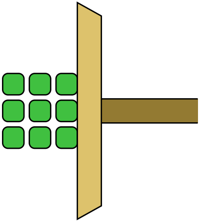
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



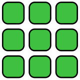
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



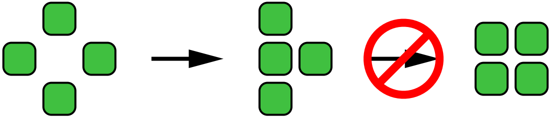
Compaction puzzle

There are 9 tokens on a grid. Can we reconfigure them into a square box by pushing them from the sides?



Compaction puzzle

There are cases where n^2 tokens cannot be pushed into a square.



Compaction puzzle: Previous results

Essentially one paper:

H. Akitaya, G. Aloupis, M. Löffler, and A. Rounds
"Trash compaction", in *Proceedings of EuroCG 2016*

Theorem

Deciding if tokens can be pushed into an $a \times b$ box is NP-complete.

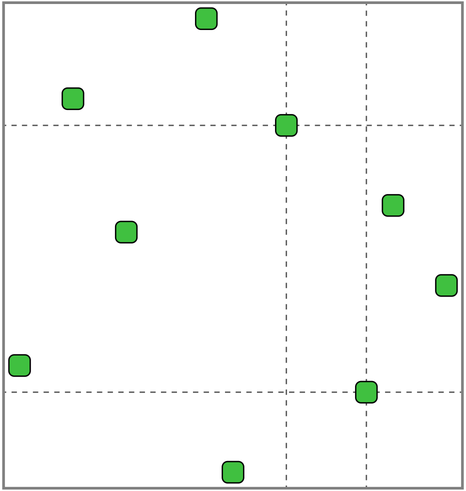
Theorem

Deciding if $2n$ tokens occupying k rows can be pushed into an $n \times 2$ box takes $O(n)$ time if $k \leq 3$ and $O(n^{2(k-2)})$ time if $k > 3$.

In our research, we study two variations of the original puzzle:

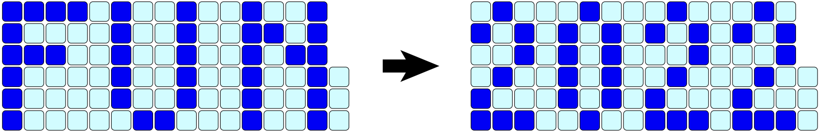
- **Compaction puzzle for sparse configurations**
- **Permutation puzzle**

Sparse compaction puzzle



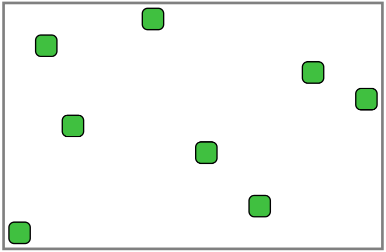
Puzzle 1. What if the configuration is *"sparse"*, i.e., no row or column has more than one token?

Permutation puzzle



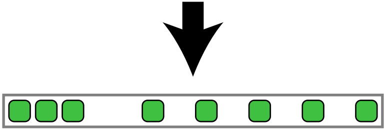
Puzzle 2. What if the configuration is already "compact", and we want to rearrange the tokens?

Pushing sparse configurations into boxes



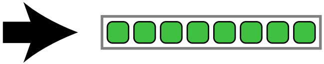
Forming an $n \times 1$ box is easy: simply push down and then right.

Pushing sparse configurations into boxes



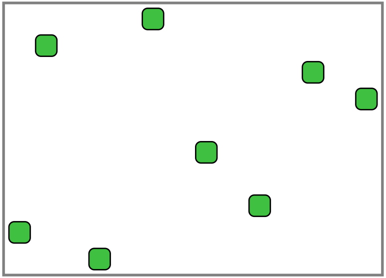
Forming an $n \times 1$ box is easy: simply push down and then right.

Pushing sparse configurations into boxes



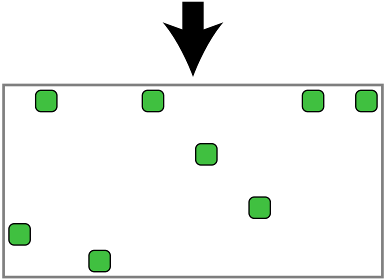
Forming an $n \times 1$ box is easy: simply push down and then right.

Pushing sparse configurations into boxes



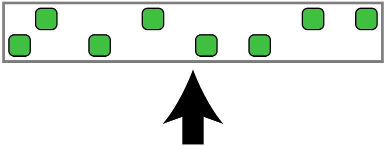
Given $2n$ tokens, forming an $n \times 2$ box is easy, too.

Pushing sparse configurations into boxes



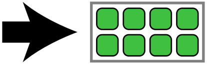
Given $2n$ tokens, forming an $n \times 2$ box is easy, too.

Pushing sparse configurations into boxes



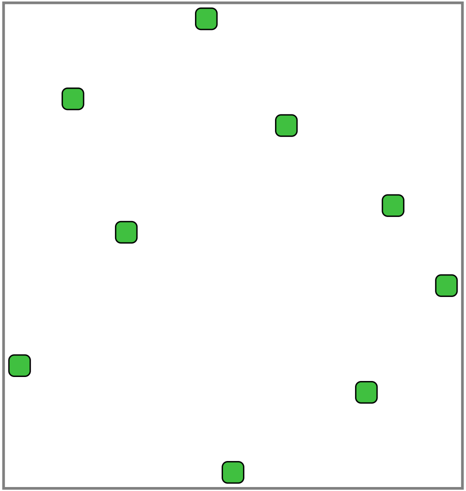
Given $2n$ tokens, forming an $n \times 2$ box is easy, too.

Pushing sparse configurations into boxes



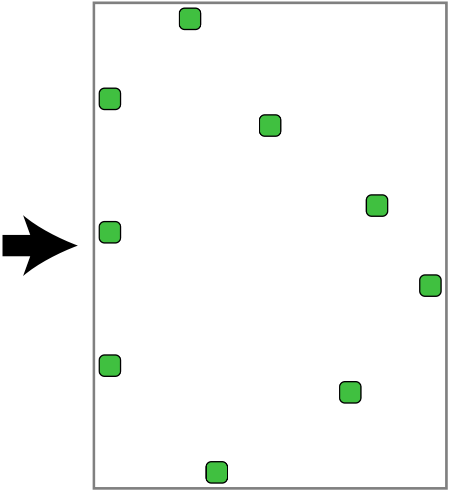
Given $2n$ tokens, forming an $n \times 2$ box is easy, too.

Pushing sparse configurations into boxes



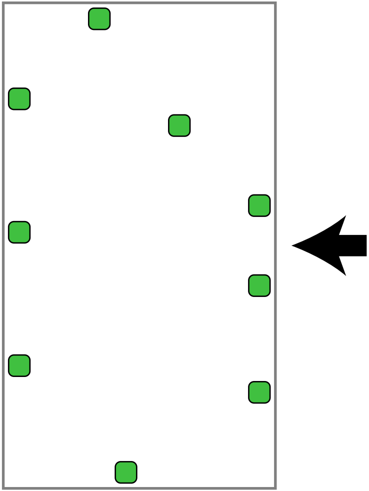
There is an algorithm for pushing 9 tokens into a 3×3 square.

Pushing sparse configurations into boxes



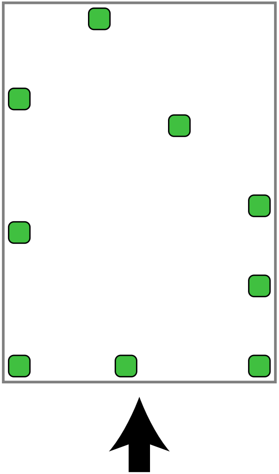
Step 1: Push right until 3 “side tokens” become aligned.

Pushing sparse configurations into boxes



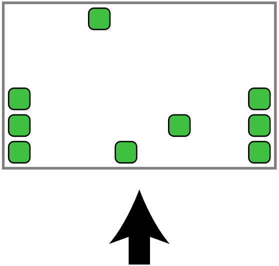
Step 2: Push left until 3 “side tokens” become aligned.

Pushing sparse configurations into boxes



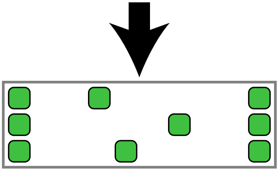
Step 3: Push up until 2 “middle tokens” are in adjacent rows.

Pushing sparse configurations into boxes



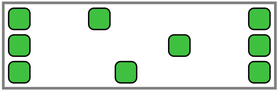
Step 3: Push up until 2 “middle tokens” are in adjacent rows.

Pushing sparse configurations into boxes



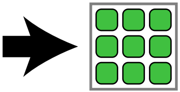
Step 4: Push down until all “middle tokens” are in adjacent rows.

Pushing sparse configurations into boxes



Now the tokens occupy exactly 3 consecutive rows.

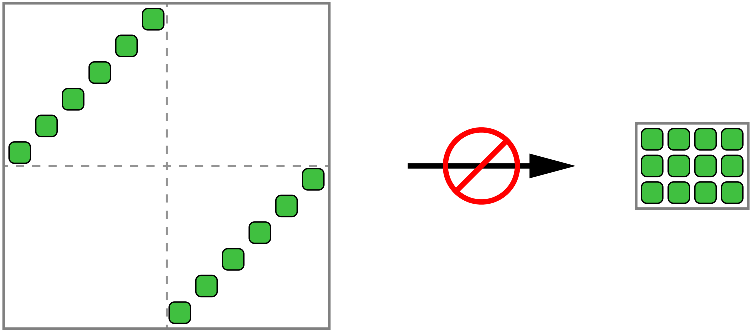
Pushing sparse configurations into boxes



Step 5: Push all the way to the left to form a 3×3 square.

Pushing sparse configurations into boxes

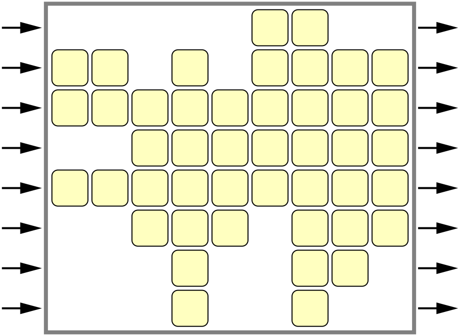
In all other cases, i.e., when $a \geq 4$ and $b \geq 3$, there are sparse configurations of $n = ab$ tokens that cannot form an $a \times b$ box:



Theorem

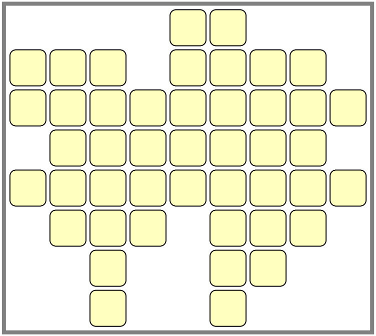
All sparse configurations of $n = ab$ tokens can be pushed into an $a \times b$ box if and only if $a \leq 2$ or $b \leq 2$ or $a = b = 3$.

Compact configurations



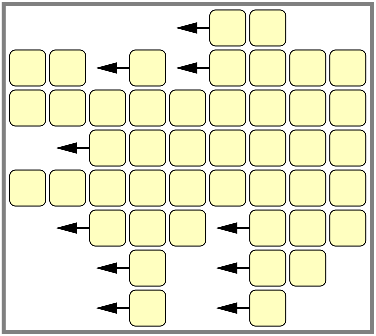
Pushing the tokens in one direction is equivalent to letting them "fall" in the opposite direction within the bounding box.

Compact configurations



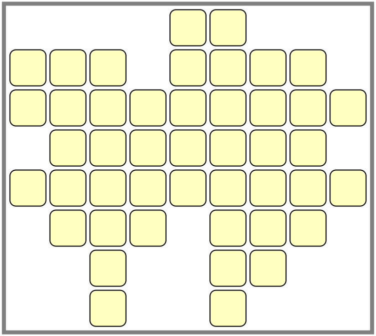
Pushing the tokens in one direction is equivalent to letting them "fall" in the opposite direction within the bounding box.

Compact configurations



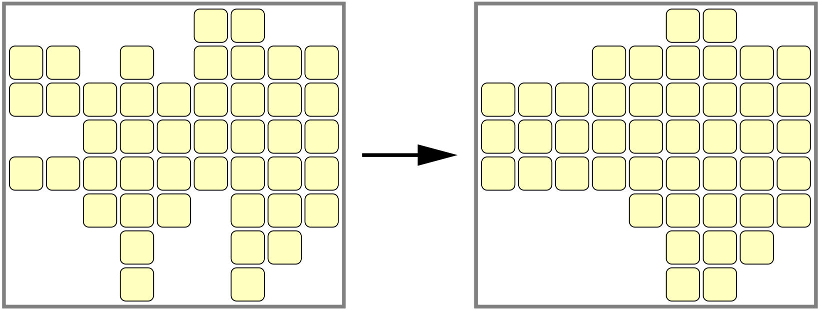
Pushing the tokens in one direction is equivalent to letting them "fall" in the opposite direction within the bounding box.

Compact configurations



Pushing the tokens in one direction is equivalent to letting them "fall" in the opposite direction within the bounding box.

Compact configurations

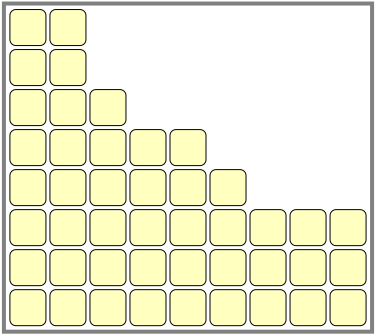


After some moves, the configuration tends to become orthogonally convex, where each row (resp. column) of length k is contained in the projection of every row (resp. column) of length $\geq k$.
We call such a configuration *compact*.

Canonical configurations

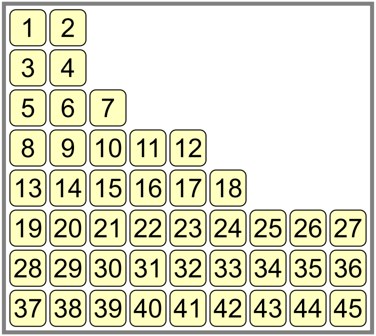
Lemma

Any move performed from a compact configuration is reversible.



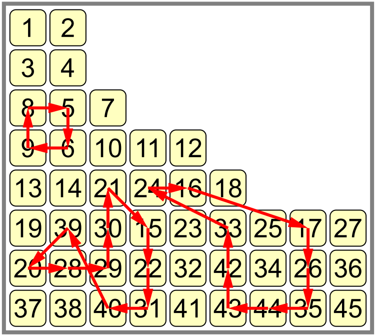
Thus, without loss of generality, we will assume the configuration to be a compact “staircase” with a full leftmost column and a full bottommost column. We call such a configuration *canonical*.

Permutation puzzle



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

Permutation puzzle

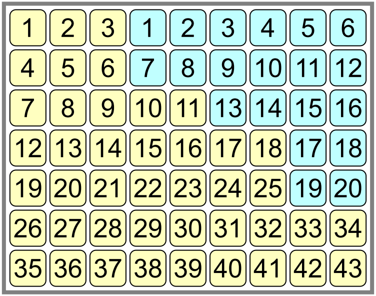


Problem: What are the possible configurations of this puzzle?

Note: The set of possible permutations is closed under composition, and therefore is a *permutation group*.*

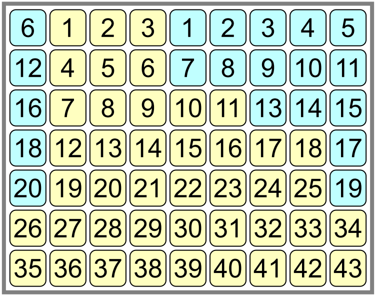
*For *finite* sets of permutations, closure under composition is sufficient.

Permutation puzzle: Permutations are even



We will prove that any permutation obtainable in this puzzle must be <u>even</u>. It will be convenient to label the empty cells, too.

Permutation puzzle: Permutations are even



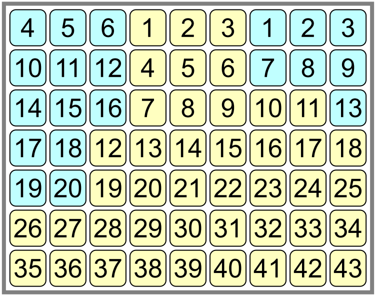
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutation puzzle: Permutations are even

5	6	1	2	3	1	2	3	4
11	12	4	5	6	7	8	9	10
15	16	7	8	9	10	11	13	14
17	18	12	13	14	15	16	17	18
19	20	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

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

Permutation puzzle: Permutations are even



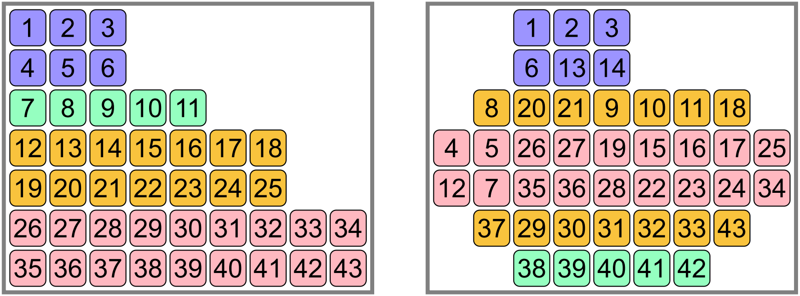
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutation puzzle: 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

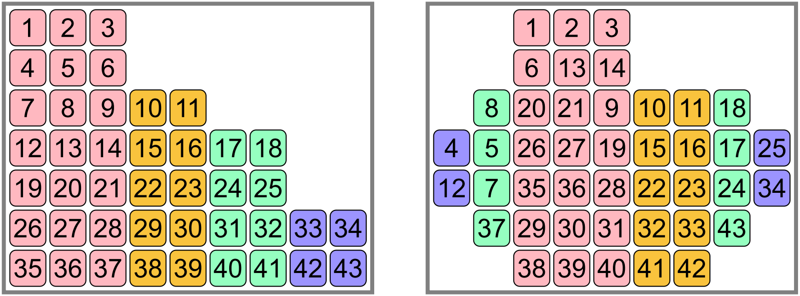
Every move causes a cyclic permutation on some rows or columns, involving both labeled tokens and labeled empty cells.

Permutation puzzle: Permutations are even



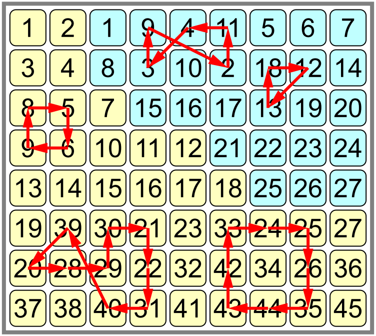
Observe that all rows of the same length must always be aligned. Moreover, for every move that pushes all rows of some length to the right, there must be a move that pushes all of them to the left.

Permutation puzzle: Permutations are even



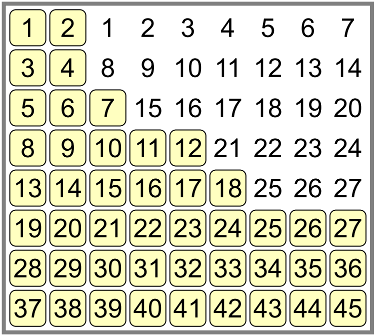
The same holds for columns and up/down moves. Thus, if we decompose every move into cycles (involving both tokens and empty cells), we see that every cycle must have a matching cycle of the same length.

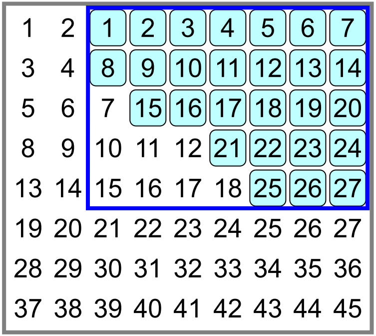
Permutation puzzle: Permutations are even



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.

Permutation puzzle: Primal and dual puzzles





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

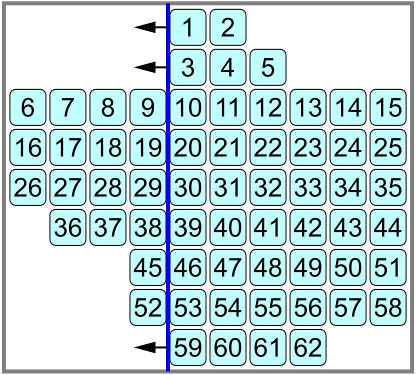
Permutation puzzle: Primal and dual puzzles

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

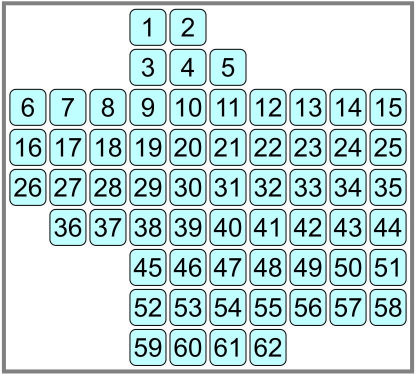
As we play the primal puzzle, we are also playing the dual puzzle.

Permutation puzzle: Primal and dual puzzles



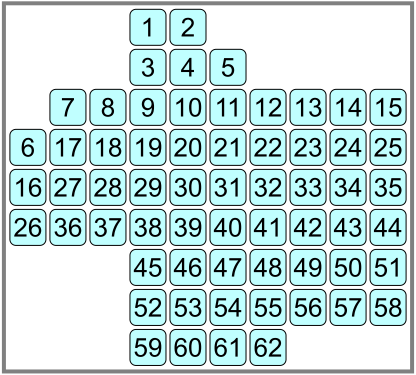
The rules of the dual puzzle as slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Permutation puzzle: Primal and dual puzzles



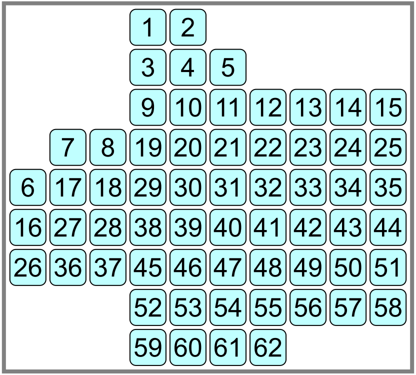
The rules of the dual puzzle as slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Permutation puzzle: Primal and dual puzzles



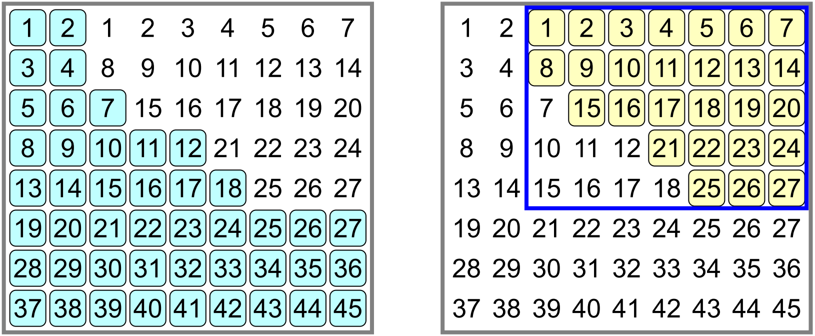
The rules of the dual puzzle as slightly different: in a left move, only the rows that are farthest from the left margin are pushed, etc.

Permutation puzzle: Primal and dual puzzles



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

Permutation puzzle: Primal and dual puzzles



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

Permutation 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.

Permutation 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

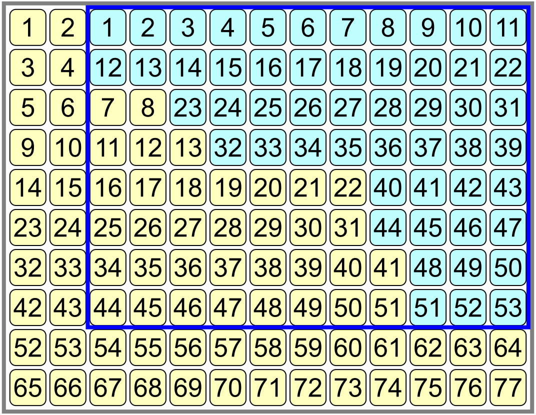
The dual puzzle of a dual-type puzzle is again a primal-type puzzle.

Permutation 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

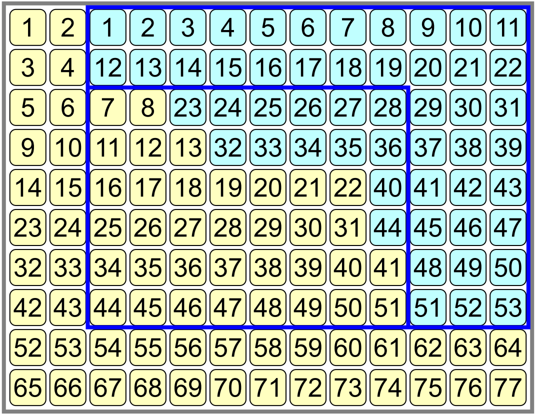
Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutation puzzle: Permutations are even



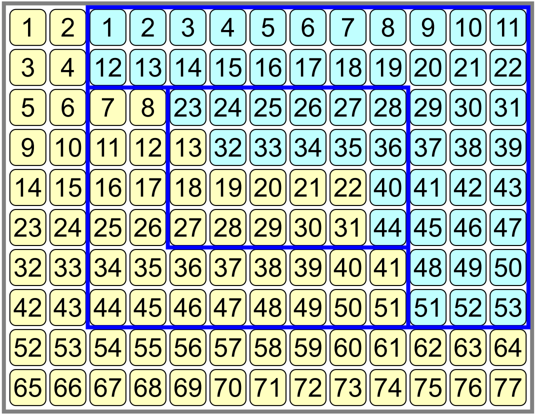
Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutation puzzle: Permutations are even

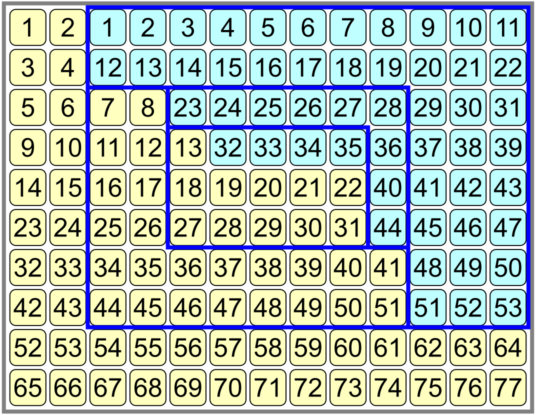


Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

Permutation puzzle: Permutations are even

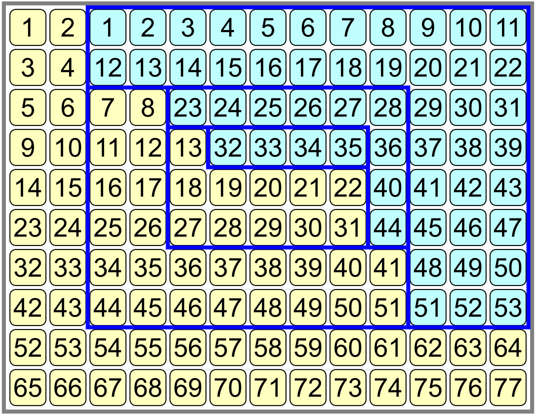


Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.



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

Permutation puzzle: Permutations are even



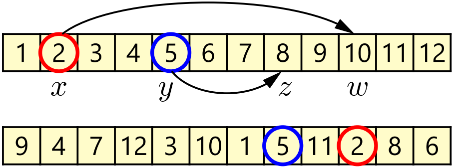
Since the dual puzzle is smaller than the primal, we can conclude by induction that the permutation restricted to each puzzle is even.

2-transitive groups

We have proved that only even permutations of the tokens are possible. Can we obtain *all* even permutations?

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 $\pi(y) = z$.



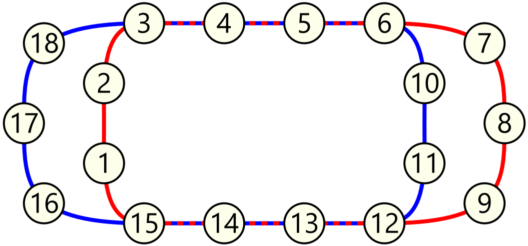
Theorem (Jones, 2014)

If a 2-transitive permutation group G on n items contains a cycle of length $n - 3$ or less, then G contains all even permutations.

2-transitive groups

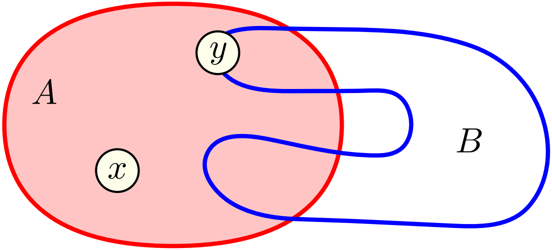
Lemma

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

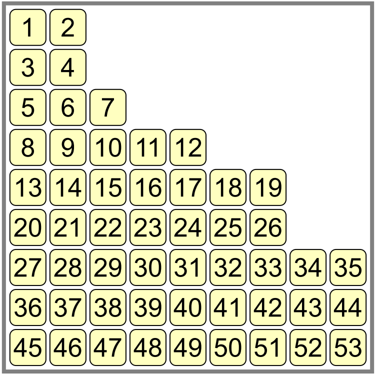


Lemma

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 $A \cup B$.

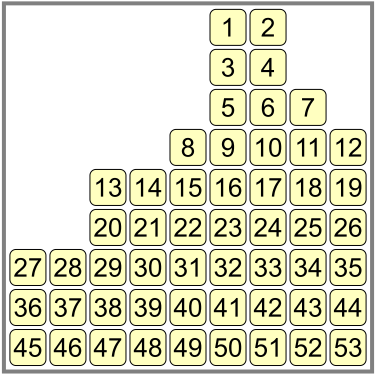


Permutation puzzle: Generating cycles



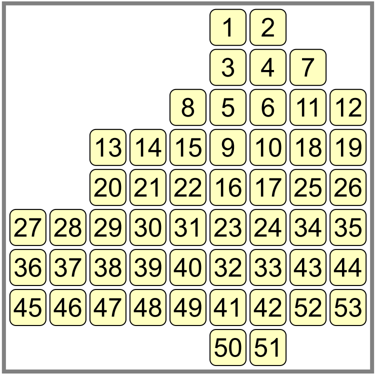
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



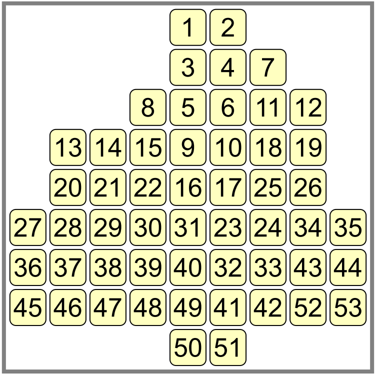
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



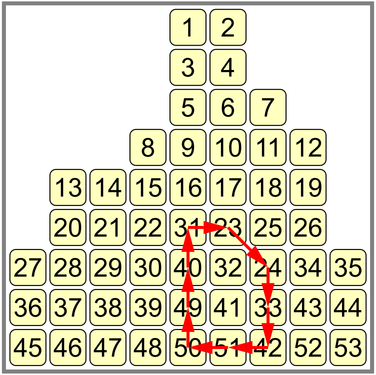
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



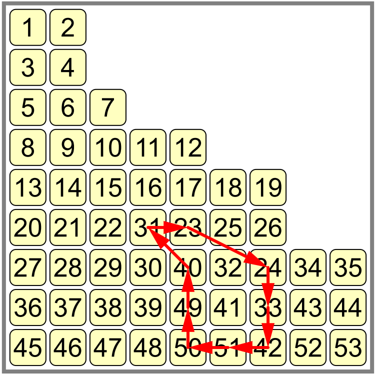
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



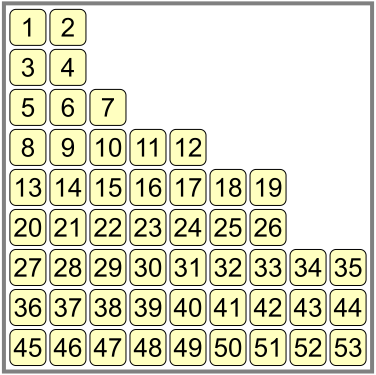
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



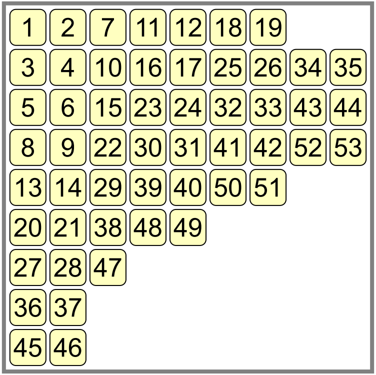
The sequence of moves $\mathsf{R}^k\mathsf{RULDL}^k$, with $k \geq 0$, generates a family of cycles spanning all the full rows.

Permutation puzzle: Generating cycles



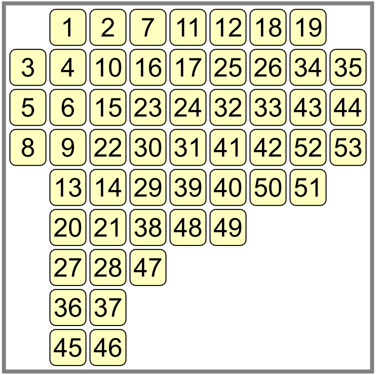
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Permutation puzzle: Generating cycles



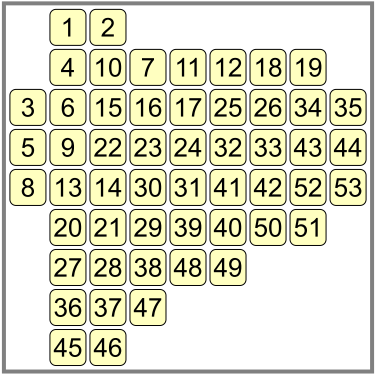
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Permutation puzzle: Generating cycles



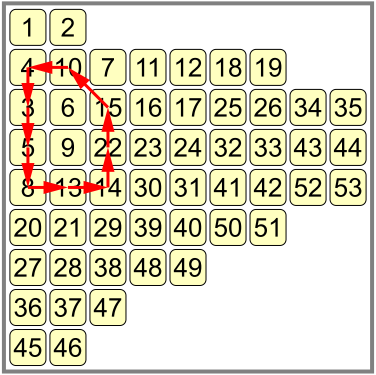
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Permutation puzzle: Generating cycles

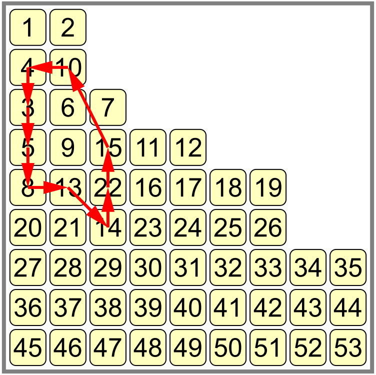


The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Permutation puzzle: Generating cycles

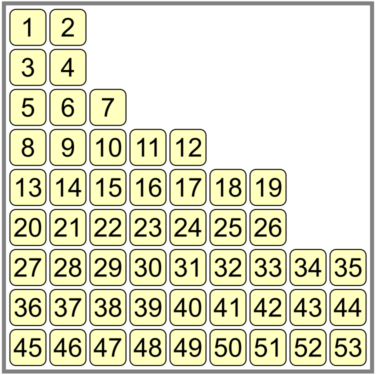


The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.



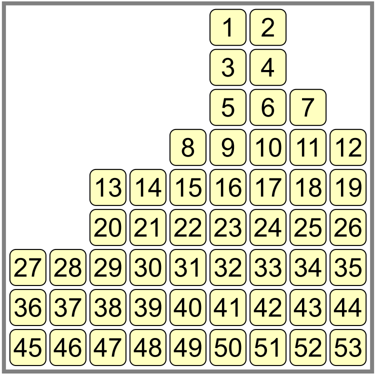
The sequence of moves $\mathsf{U}^k\mathsf{URDLD}^k$, with $k \geq 0$, generates a family of cycles spanning all the full columns.

Permutation puzzle: Generating cycles



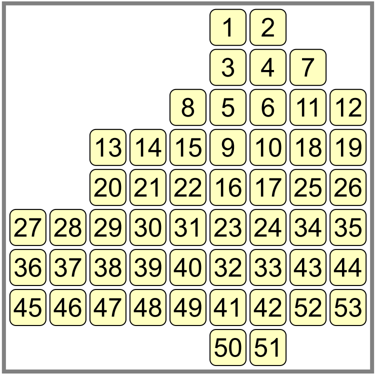
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



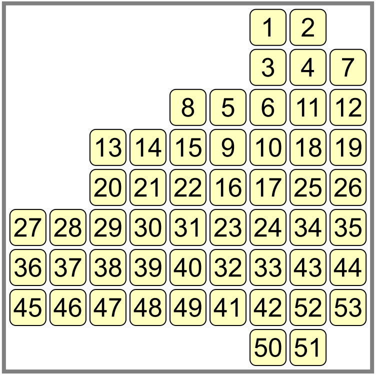
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



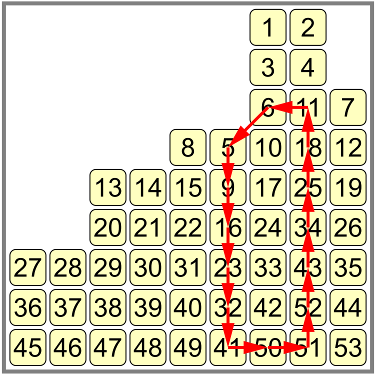
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



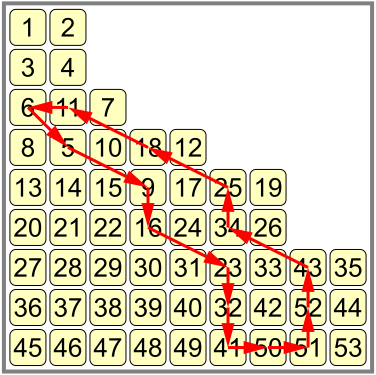
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



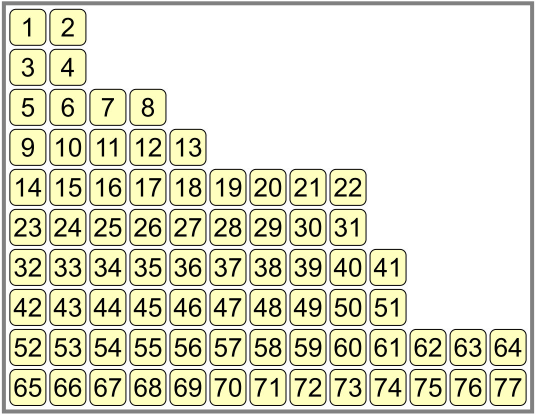
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



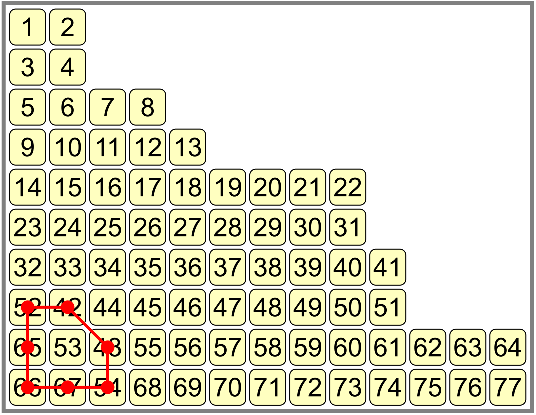
The sequence of moves $\mathsf{R}^k\mathsf{URDLL}^k$, with $k \geq 0$, generates a family of cycles spanning all the non-full rows and columns.

Permutation puzzle: Generating cycles



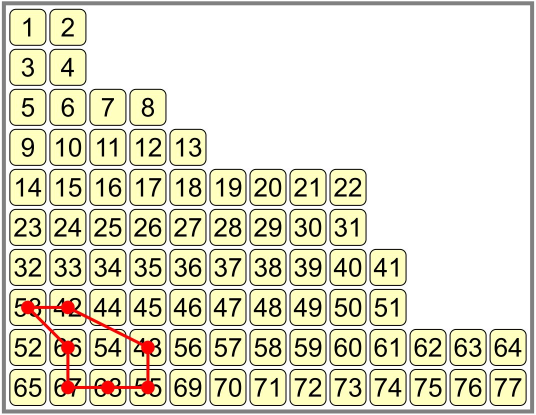
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



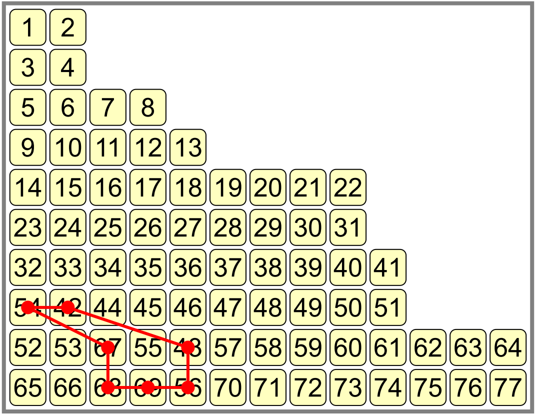
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



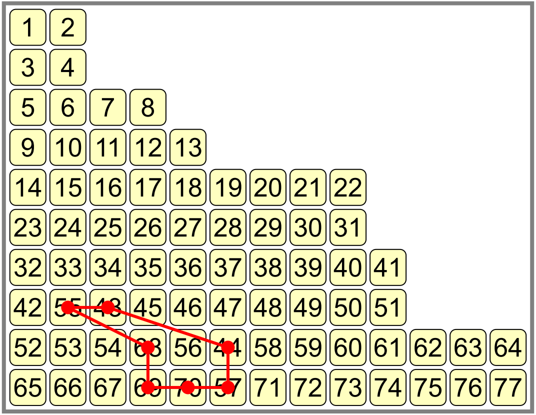
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



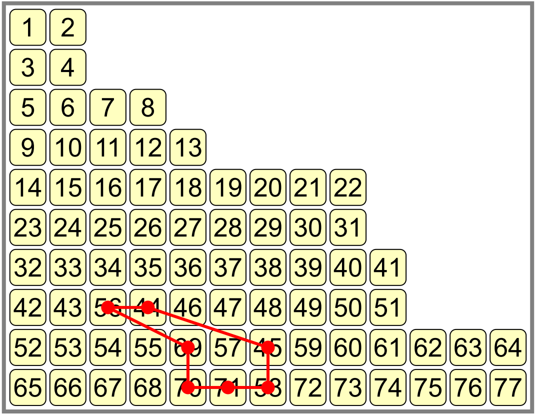
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



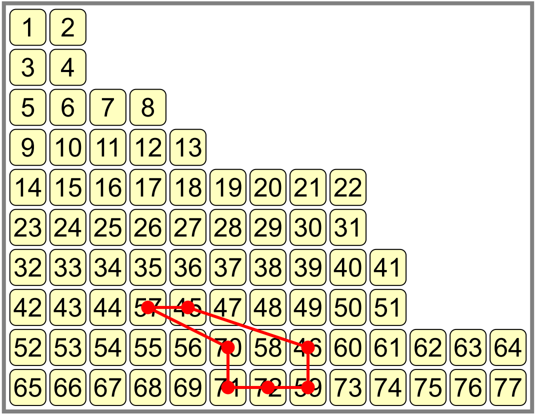
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



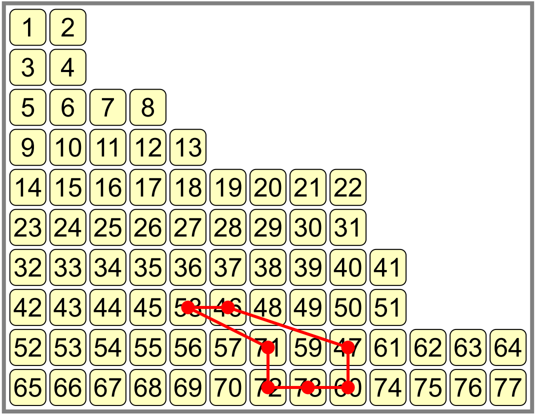
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



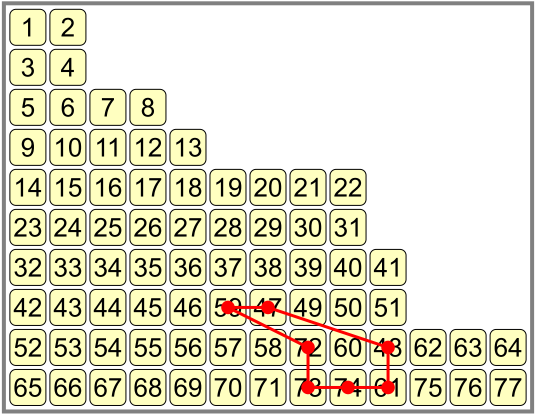
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



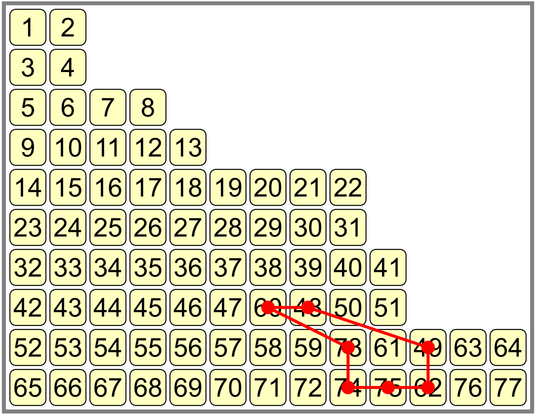
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



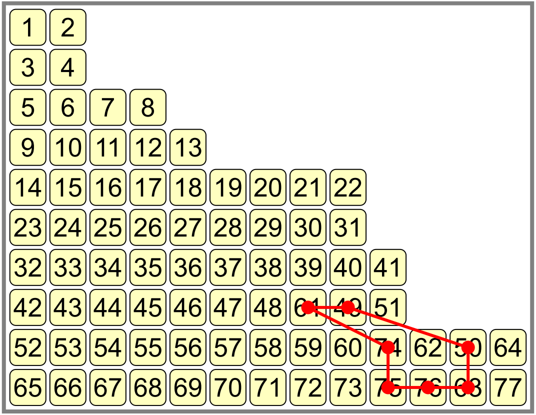
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



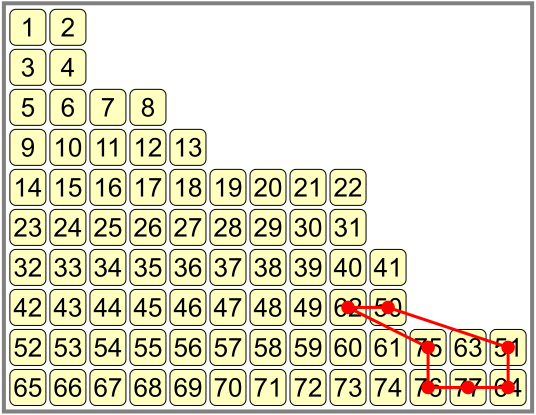
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



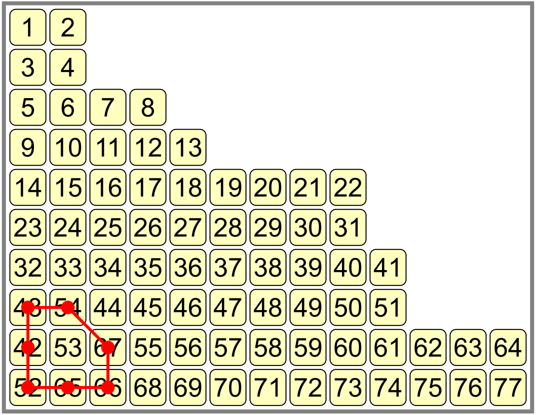
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



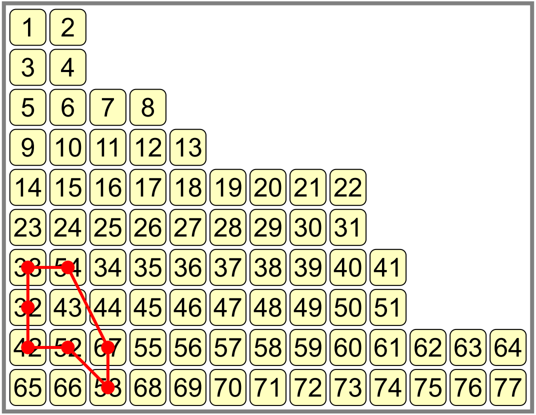
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



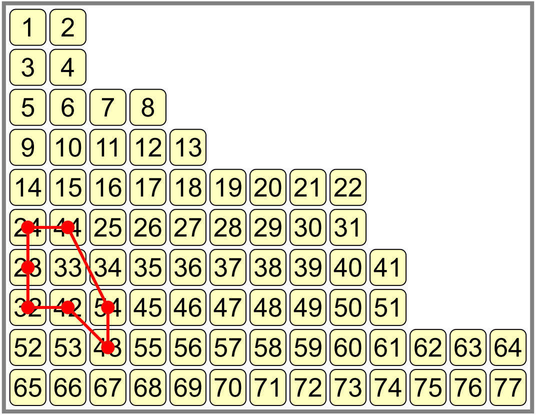
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



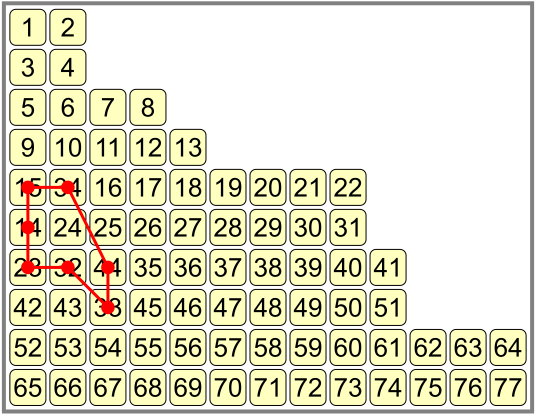
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



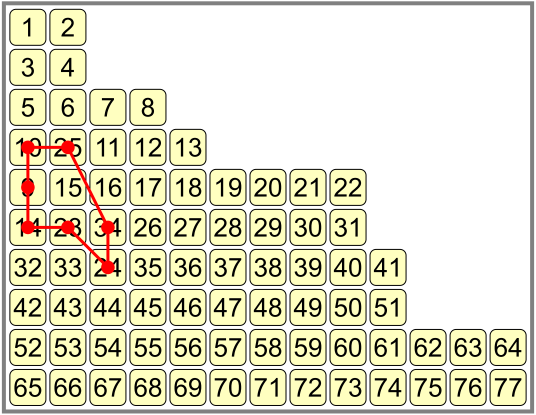
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



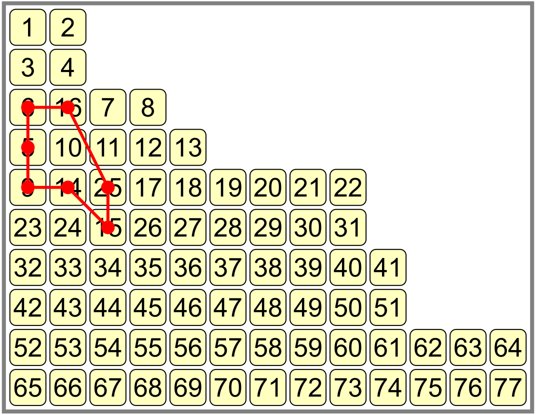
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



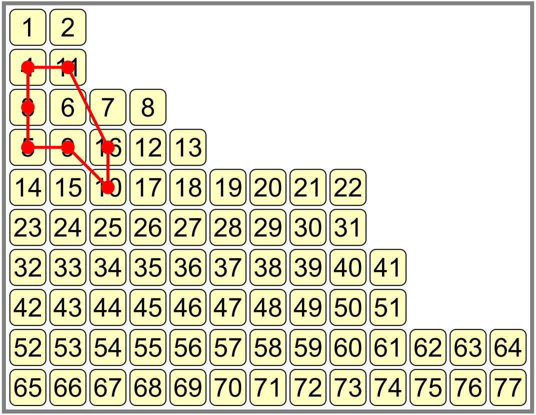
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



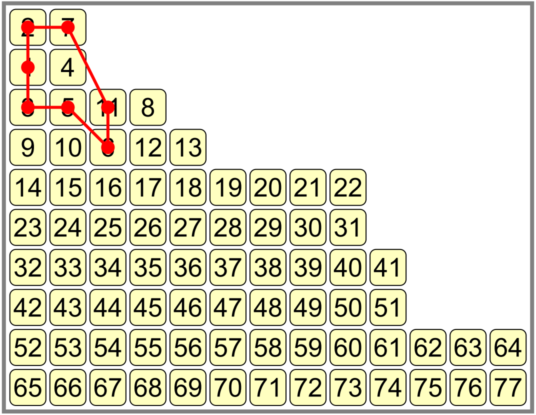
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



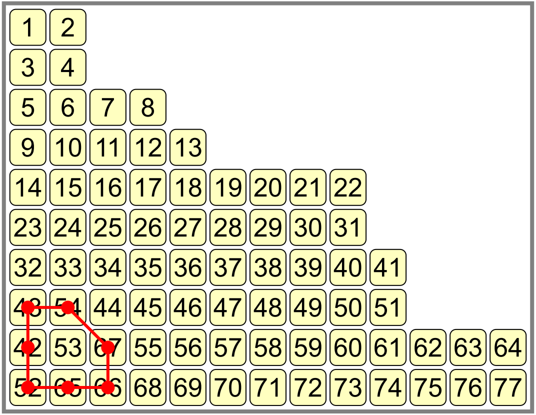
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



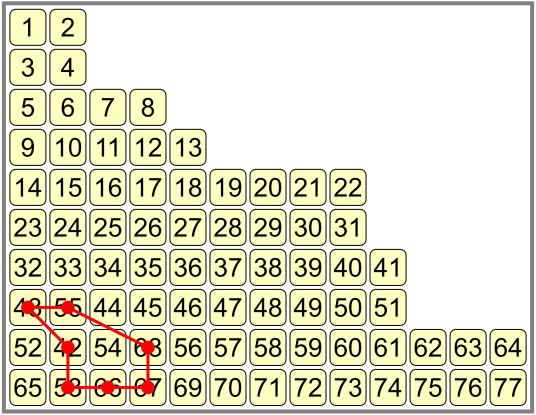
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



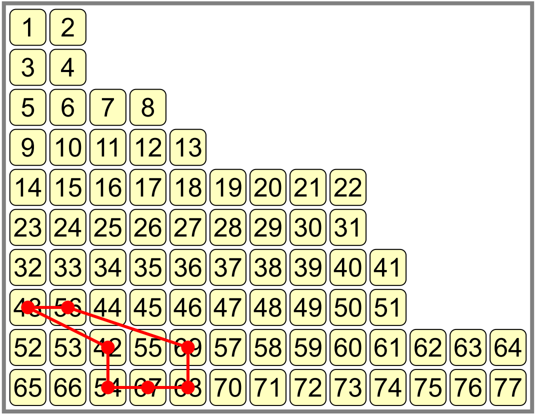
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



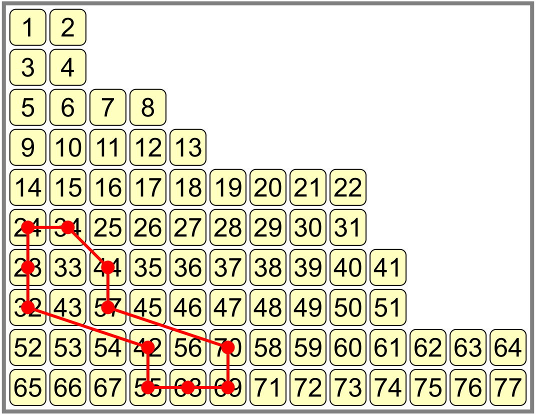
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



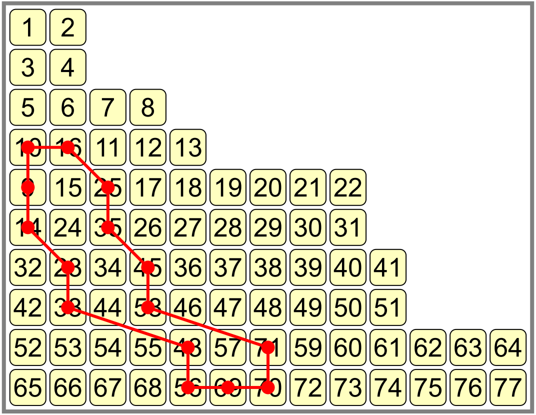
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



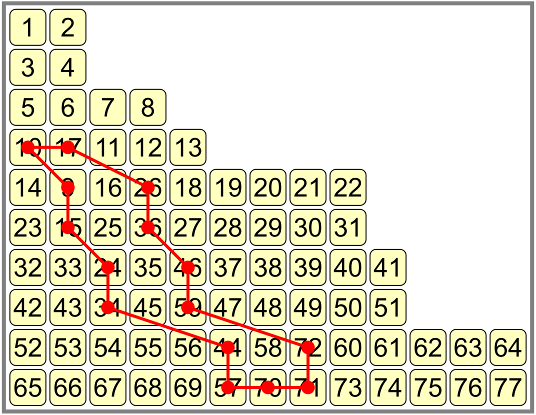
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



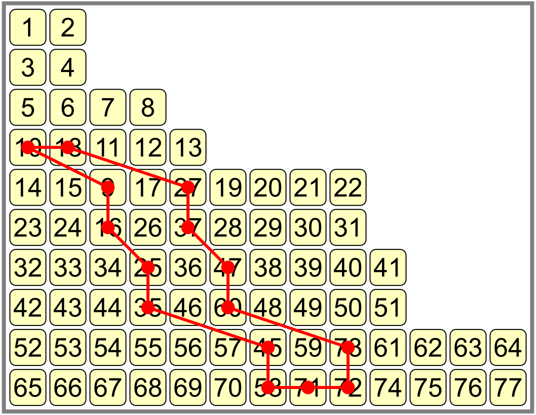
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



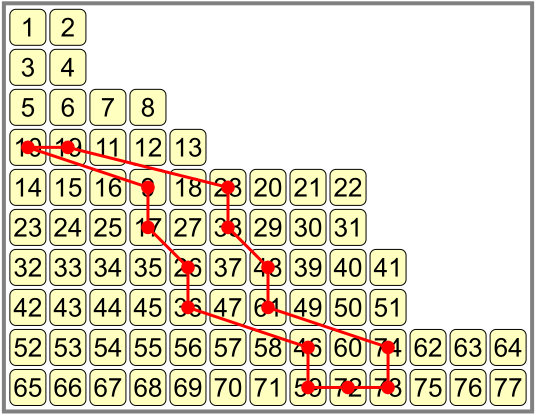
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



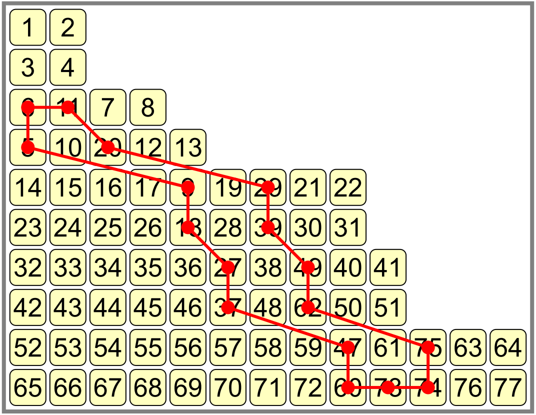
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



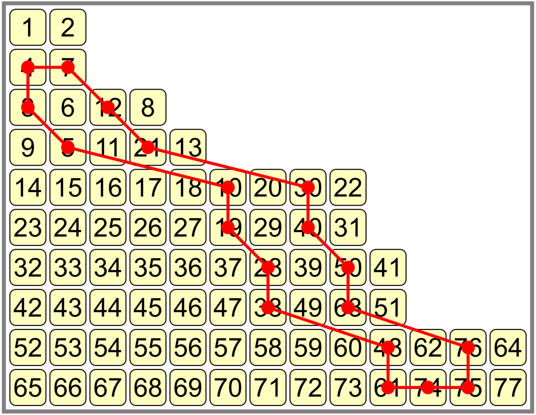
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



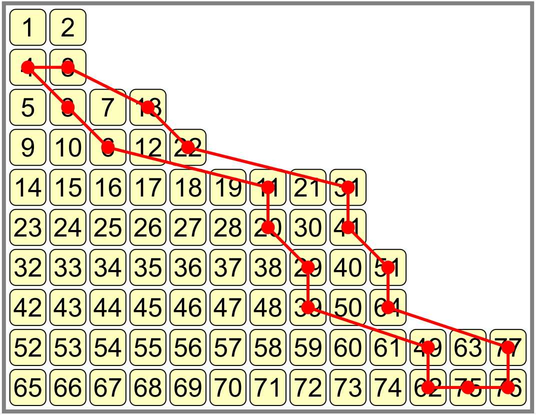
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



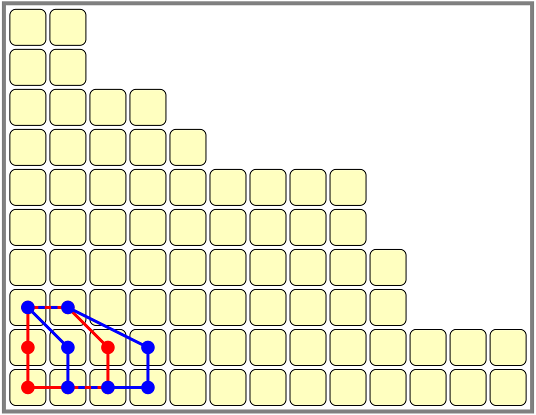
These cycles collectively span all tokens.

Permutation puzzle: Generating cycles



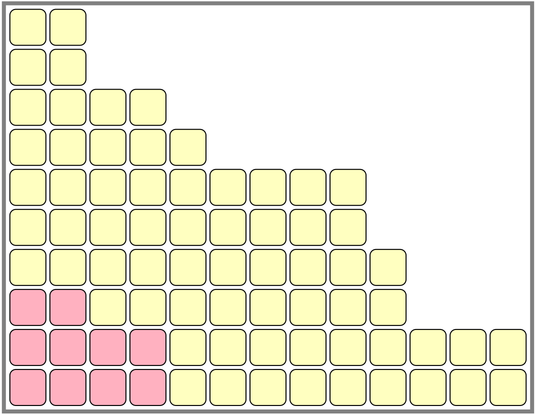
These cycles collectively span all tokens.

Permutation puzzle: Generating all even permutations



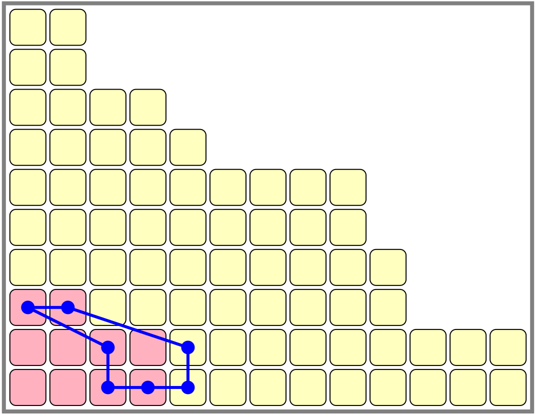
By our first lemma, these two cycles determine a 2-transitive permutation group acting on the tokens they span.

Permutation puzzle: Generating all even permutations



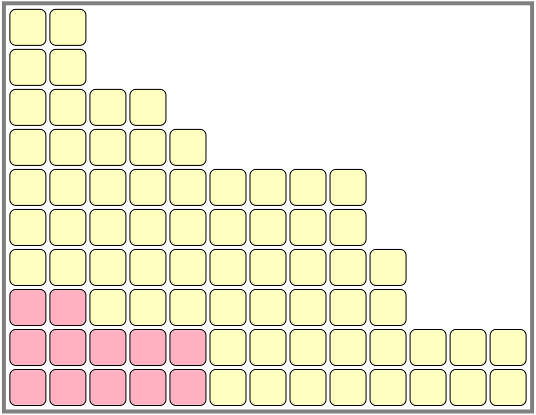
By our first lemma, these two cycles determine a 2-transitive permutation group acting on the tokens they span.

Permutation puzzle: Generating all even permutations



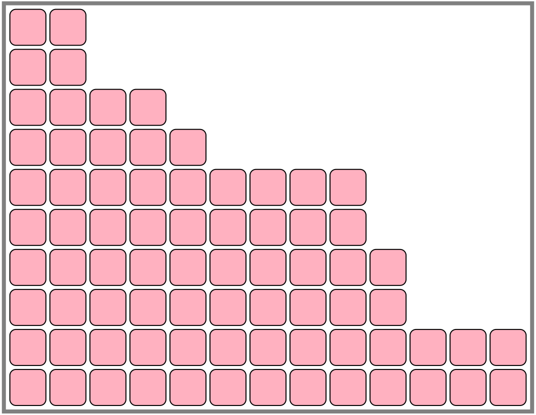
By our second lemma, we can extend this set to the tokens spanned by the next cycle.

Permutation puzzle: Generating all even permutations



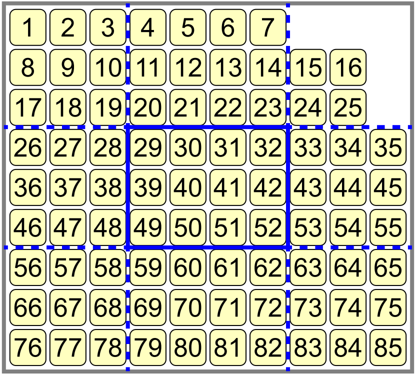
Proceeding in this fashion, we conclude that our cycles act 2-transitively on all the tokens they span.

Permutation puzzle: Generating all even permutations



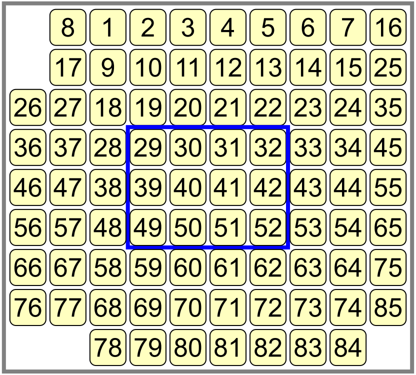
Thus, by Jones' theorem, we can generate all even permutations on the tokens spanned by our cycles.

Permutation puzzle: Unmovable core



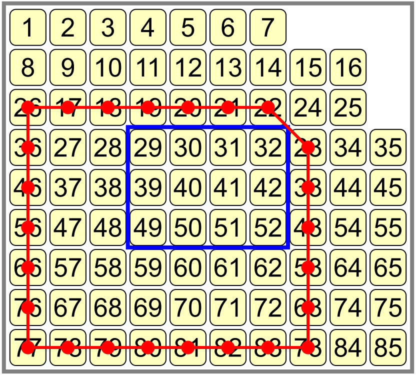
In general, if more than half of the rows and more than half of the columns are full, there is a *central core* that is impossible to move.

Permutation puzzle: Unmovable core



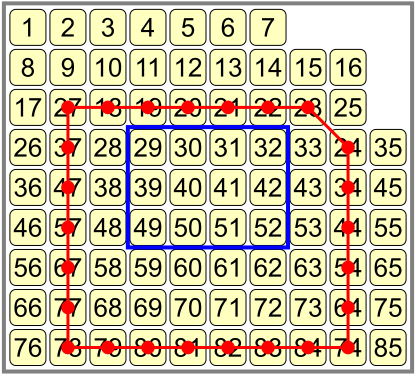
In general, if more than half of the rows and more than half of the columns are full, there is a *central core* that is impossible to move.

Permutation puzzle: Unmovable core



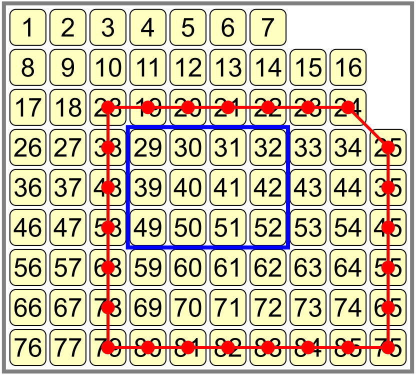
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



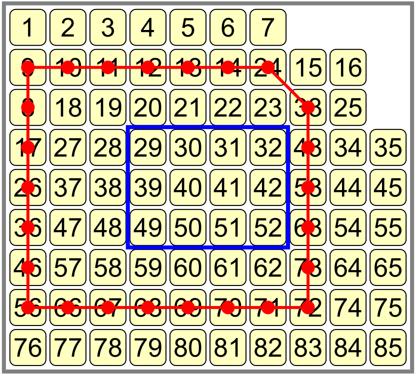
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



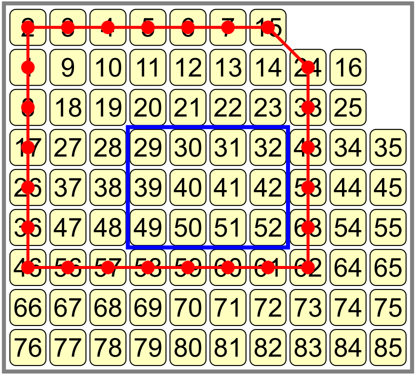
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



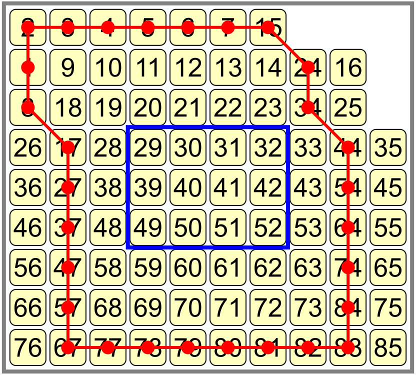
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



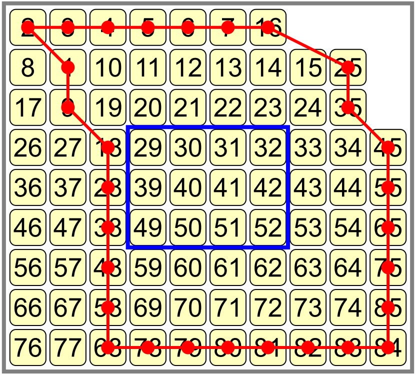
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



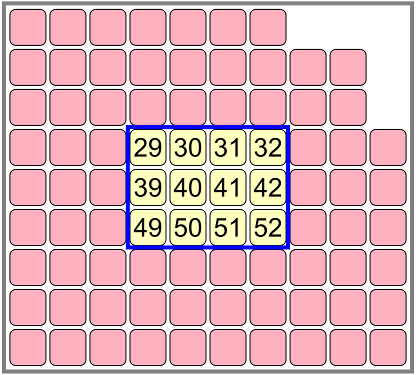
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



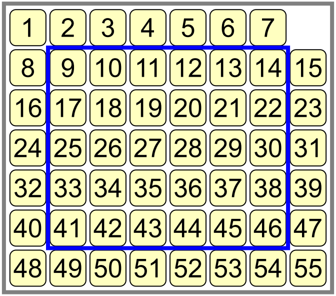
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Unmovable core



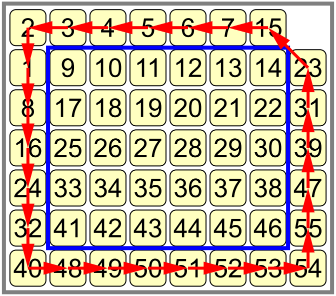
However, there are cycles spanning all non-core tokens. Thus, we can generate all even permutations of the non-core tokens.

Permutation puzzle: Special cases



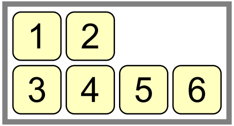
An exception is when there is exactly one empty cell. In this case, the permutation group is easily seen to be cyclic.

Permutation puzzle: Special cases



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

Permutation puzzle: Special cases

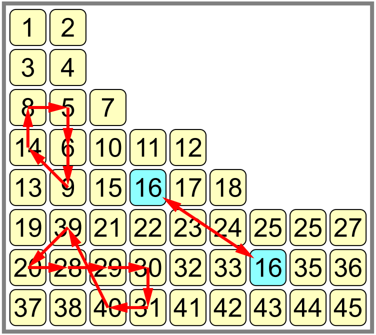


The only other exception is the puzzle with a 4×2 bounding box and two empty cells. In this case, the permutation group has order 60 and is isomorphic to the alternating group A_5 (not A_6!).*

*Trivia: S_6 is the only S_n having a transitive subgroup isomorphic to A_{n-1}.

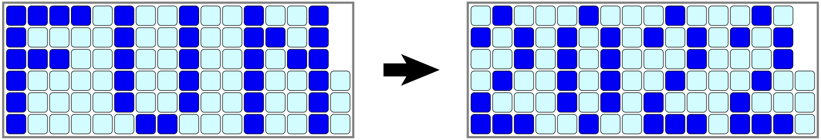
Permutation puzzle: Non-unique labels

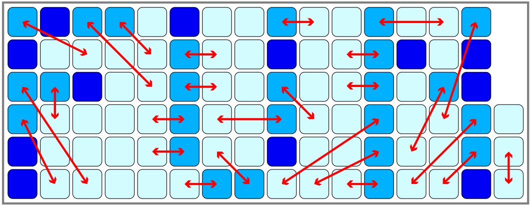
What if not all labels are unique? Note that we can transpose any two equal (non-core) labels “for free”.



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 $\pi \circ (x\ y)$.

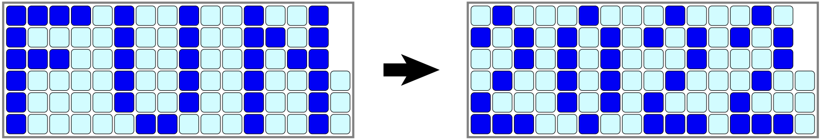
Permutation puzzle: Non-unique labels

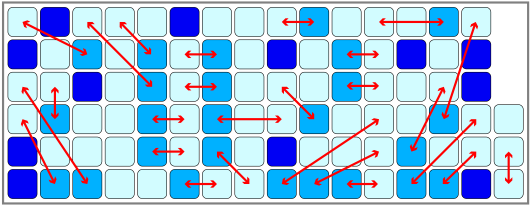




In this puzzle, since the core is empty and some tokens are equal, *all* permutations are possible (including the odd ones).

Permutation puzzle: Non-unique labels





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

Conclusion and open problems

Theorem (Sparse compaction puzzle)

All sparse configurations of $n = ab$ tokens can be pushed into an $a \times b$ box if and only if $a \leq 2$ or $b \leq 2$ or $a = b = 3$.

Open problem

Is it NP-complete to decide whether a given sparse configuration can be pushed into a given rectangle?

Theorem (Permutation puzzle)

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.*

Open problem

What if the tokens form a non-compact configuration?