The exact diameter of the symmetric group S_n with LRX generators

S. Uskov[1], A. Chervov[2], F. Levkovich-Maslyuk[3], I. Litvak[4], A. Smolensky[5], A. Soibelman[6], and D. Taha[7]

[1] *Independent researcher,* specm32 • gmail.com
[2] *Institute Imagine, Paris, France,* al.chervov • gmail.com
[3] *Centre for Mathematical Science, City St George's, University of London, Northampton Square, EC1V 0HB, London, UK,* fedor.levkovich • gmail.com
[4] *Independent researcher,* Vanya.litvak • gmail.com
[5] *Neapolis University Pafos, Cyprus,* andrei.smolensky • gmail.com
[6] *IHES, France,* asoibelman • gmail.com
[7] *Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany,* diaaeldin.taha • mis.mpg.de

hexagon:2611.00333v1

Abstract

We consider the undirected Cayley graph of the symmetric group S_n with respect to the cyclic shift L, its inverse R, and the transposition X of the first two positions. We prove that its diameter is given by $n(n-1)/2$ for every $n \geq 4$, thus settling a conjecture open since at least 2011. This is done by separately establishing sharp upper and lower bounds, invoking a variety of combinatorial and group-theoretic arguments which leave a finite set of special cases that are verified in the accompanying C++ code. We also provide a Lean formalization of the results.

Contents

1 Introduction **2**
1.1 Related work . . . 3
1.2 Examples for small n . . . 4
1.3 Organization of the paper . . . 5
1.4 AI statement and agentic system . . . 6

2 Strategy of the proof **6**

3 Setup and basic definitions **7**

4 The lower bound **7**

5 The upper bound: a complete constructive proof **9**
5.1 Choosing a lift and its affine invariants . . . 9
5.2 Original root packets . . . 12
5.3 Exact words using a shared logical buffer . . . 13
5.4 A quadratic estimate for a filling . . . 15
5.5 The entire connected case . . . 16
5.6 Joining components and transferring a word . . . 17
5.7 Finite inversion and structural lemmas . . . 19
5.8 A large shallow shell . . . 22

5.9 Exactly two finite components . . . 25
5.10 At least three components: a finite cutoff . . . 26

6 The complete finite certificate 28
6.1 Scalar parameters and necessary filters . . . 28
6.2 All outside permutations and the sign tests . . . 29
6.3 Enumeration output and the remaining routes . . . 30
6.4 Reproducibility and exact arithmetic . . . 30

7 Completion of the proof and reproduction 31

8 Formalization 31

9 Acknowledgements 31

A Details of the independent lower-bound proof 31

B Proofs and identifications for standard ingredients 33
B.1 Finite inversion and transposition lengths . . . 33
B.2 Affine length and the preservation of rotations . . . 34
B.3 The median estimate used for rotations . . . 34

C Majorization notation and the lift calculation 35

D Supplementary proofs for standard results 35
D.1 A rank proof of the Bruhat transfer . . . 35
D.2 Shallow involutions and noncrossing chords . . . 35

E Complete C++17 certificate 36

1 Introduction

The problem and main results. In this paper we consider the following group-theoretic problem. Take a row of n cards labelled $0, 1, \dots, n-1$, and allow three moves:

- L: move the first card to the end, $(x_0, x_1, \dots, x_{n-1}) \mapsto (x_1, \dots, x_{n-1}, x_0)$;
- $R = L^{-1}$: move the last card to the front;
- X: exchange the first two cards, $(x_0, x_1, x_2, \dots) \mapsto (x_1, x_0, x_2, \dots)$.

What is the largest number of moves that may be needed to turn one arrangement into another, moving optimally?

Equivalently, in more formal terms, this problem is about the diameter of the Cayley graph of the permutation group with the three generators L, R, X. Let Γ_n be the undirected Cayley graph of S_n with respect to the cyclic shift L, its inverse R, and the transposition X of the first two positions. Its vertices are the $n!$ arrangements, and two arrangements are adjacent when a single move turns one into the other. The question asks for the *diameter* $\operatorname{diam}\Gamma_n$, the largest distance between two vertices. Throughout we put

$$N = \binom{n}{2} = \frac{n(n-1)}{2}.$$

In this work we establish the following theorem.

Theorem 1.1. *For every $n \geq 4$ the diameter of Γ_n equals N. The distance N is attained, for example, between $e = (0, 1, \ldots, n-1)$ and $p^* = (1, 0, n-1, n-2, \ldots, 2)$.*

The theorem amounts to two inequalities. Denoting the distance between elements a and b by $d(a, b)$, we have a *lower bound*, $d(e, p^*) \geq N$, and an *upper bound*, stating that every arrangement can be reached from e in at most N moves. Since Γ_n is vertex-transitive (Section 3), the upper bound gives $\operatorname{diam} \Gamma_n \leq N$.

Motivation and context. One of the motivations to attack this problem is that it can be viewed as a step towards the much more difficult task of proving a general Babai-type conjecture: a polynomial upper bound on the diameter of S_n with *any* generating set [1–5]. We hope that the present work together with rigorous proofs of diameter results for particular generators (many of which were recently conjectured in [6]) will ultimately bring this goal closer. Further interest in the kind of question we consider here comes from the fruitful idea of viewing word decomposition problems in finite groups as a reinforcement learning environment, and from recent interest in AI-assisted exploration of similar questions [7–15].

AI assistance. The proof provided in this paper is mostly AI-generated with the help of our agentic system tailored for mathematical discovery (see section 1.4). The proof is accompanied by a Lean formalization. We hope to provide a clearer exposition as well as to simplify the proof in the future.

1.1 Related work

The sequence and the conjecture. The diameters $\operatorname{diam} \Gamma_n$ form the OEIS sequence A186783 [16], created by T. Bartoletti in 2011. Its terms are listed there up to $n = 13$:

$$0,\ 1,\ 2,\ 6,\ 10,\ 15,\ 21,\ 28,\ 36,\ 45,\ 55,\ 66,\ 78 \qquad (1 \leq n \leq 13).$$

The conjecture $\operatorname{diam} \Gamma_n = \binom{n}{2}$ for $n > 3$ was recorded in the entry by P. W. Alexandersson in 2020 [16]. The entry also records the conjecture that the farthest element is unique and equals $(1\,2)(3\,n)(4\,n{-}1)\cdots$ in one-based cycle notation, which is our p^*.

Computational approach and earlier bounds. In [17] Chervov et al. use Γ_n as a benchmark for machine-learning pathfinding on large Cayley graphs. They give computational support for the conjecture, identify the conjectured farthest element, and produce a word for it of length $\binom{n}{2}$, so that $d(e, p^*) \leq \binom{n}{2}$. They also rigorously prove weaker bounds:

$$\frac{n(n-1)}{2} - \frac{n}{2} \ \leq\ \operatorname{diam} \Gamma_n \ \leq\ \frac{n(n-1)}{2} + 3n.$$

Our theorem 1.1 closes the gaps (which are of order n) between these bounds and the actual diameter.

The sharp lower bound. Antiufeev [18, Theorem 2] bounds from below the word length of the n elements $s\,(1\,n\cdots 2)^i$, $1 \leq i \leq n$, where s is the full reversal. These are the cyclic rotations of the reversed arrangement $(n-1, n-2, \ldots, 0)$, and p^* is one of them. From this he proposed a proof of the sharp lower bound $\operatorname{diam} \Gamma_n \geq \binom{n}{2}$. As that proof has not been verified in Lean, we prove the lower bound independently in Section 4 and Appendix A, without using results of that paper. Our argument only concerns the single element p^*, which suffices for the diameter.

Directed LX case. A closely related problem is establishing the diameter of the same Cayley graph for the case with only L and X generators – viewing it now as a directed graph. This question, open since the 1960s, goes back to Glushkov [19], who raised it in connection with the completeness of systems of operations in digital computers. Pogorelov obtained quadratic bounds [20], and Zubov established the asymptotics $\operatorname{diam}\vec{\Gamma}_n \sim \frac{3}{4}n^2$, where $\vec{\Gamma}_n$ denotes the directed Cayley graph of S_n with respect to $\{L, X\}$ (see the survey [21] for these and related results). The exact values form the OEIS sequence A039745 [22]. Exhaustive computations up to $n = 15$ led Chervov et al. to conjecture an exact formula [6, Conjecture 11]. In our recent work [23] we proved this conjecture rigorously with AI-assisted tools, in particular using the same agentic system that provided the LRX proof in the present paper.

Classical ingredients. Our proof also uses a number of classical combinatorial results relevant to this problem. Let us briefly review them here. First, if any two adjacent cards could be exchanged at unit cost, the problem would be sorting by adjacent transpositions, where the length of a permutation π of b elements equals its inversion number [24, Proposition 1.5.2],

$$I(\pi) = \#\{(i, j) : 0 \le i < j < b,\ \pi(i) > \pi(j)\}.$$

In our case, due to availability of the cyclic shifts, exchanges are naturally counted in the affine symmetric group of type $\widetilde{A}_{n-1}$, whose length function is given by an explicit inversion formula [25]. The general theory of Coxeter groups [24] provides the deletion property and Bruhat order we use in the upper bound.

The rotations are controlled by the total displacement of the cards. For finite permutations this corresponds to the quantity known as Spearman's footrule, which satisfies the Diaconis–Graham inequality [26, Theorem 2]:

$$D(\pi) = \sum_i |\pi(i) - i|, \qquad I(\pi) + \mathrm{rk}(\pi) \le D(\pi),$$

where $\mathrm{rk}(\pi)$ is the minimal number of arbitrary transpositions realising a given permutation. The footrule is twice the depth of a permutation, studied by Petersen and Tenner [27], and permutations attaining equality in the Diaconis–Graham lower bound are the shallow permutations characterized by Berman and Tenner [28]. We prove the affine analogue of this inequality (needed to prove our upper bound), and use the structure of shallow involutions in the extremal cases.

Two further classical facts enter. The constraints satisfied by the energy-minimizing lift (explained in detail in the corresponding section) form a majorization condition, which Rado's theorem relates to the permutohedron [29]. The two-colour rule we use in the lower bound is an instance of Harary's structure theorem for balanced signed graphs [30]; see also [31, Section 5.2].

Note added. We learned that, after our proof was completed, A. Guerrieri also independently obtained [32] a proof for the LRX diameter as well as several related conjectures from [6].

1.2 Examples for small n

For illustration let us collect a few examples at small values of n. For $n = 4$ we have the farthest element $p^* = (1, 0, 3, 2)$, and the word $XLLXRR$ of $6 = \binom{4}{2}$ letters takes 0123 to 1032; no shorter word does (Figure 1).

More generally, breadth-first search for small n gives:

n	3	4	5	6	7	8	9
$\operatorname{diam}\Gamma_n$	2	6	10	15	21	28	36
$\binom{n}{2}$	3	6	10	15	21	28	36
vertices at maximal distance from e	2	1	1	1	1	1	1

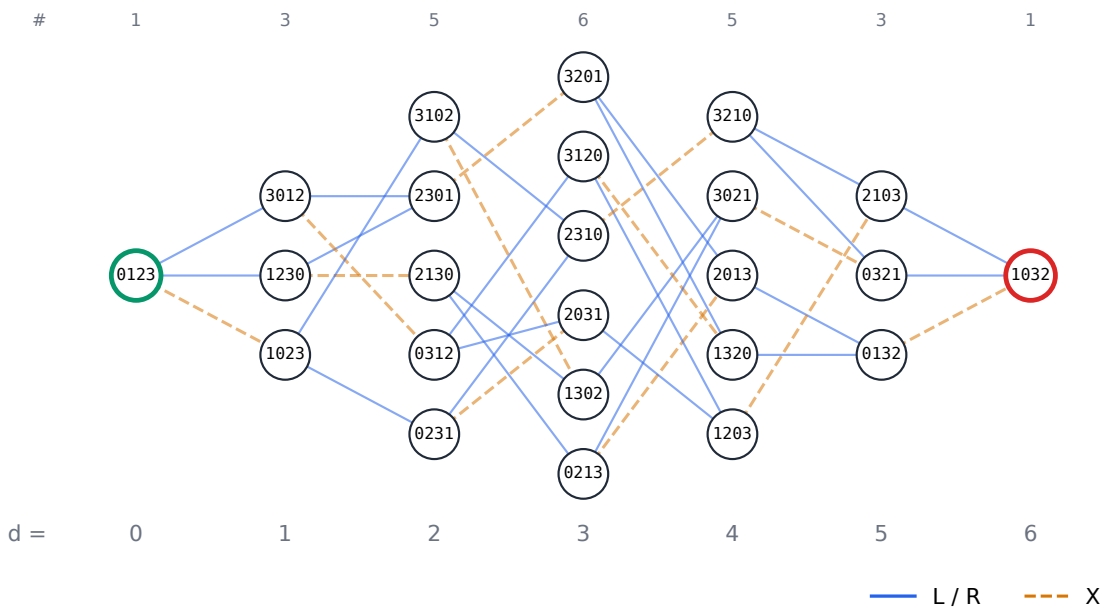


Figure 1: Γ_4 (24 vertices) arranged by distance from $e = 0123$. Solid edges: L/R; dashed: X. The layers have $1, 3, 5, 6, 5, 3, 1$ vertices; the only vertex at distance 6 is $1032 = p^*$.

For $n = 3$ the diameter formula fails (from 012 everything is reached in two moves), so the condition $n \geq 4$ is needed. For $4 \leq n \leq 9$ one can easily check explicitly that the farthest vertex from e is unique and equals p^* (Figure 2).

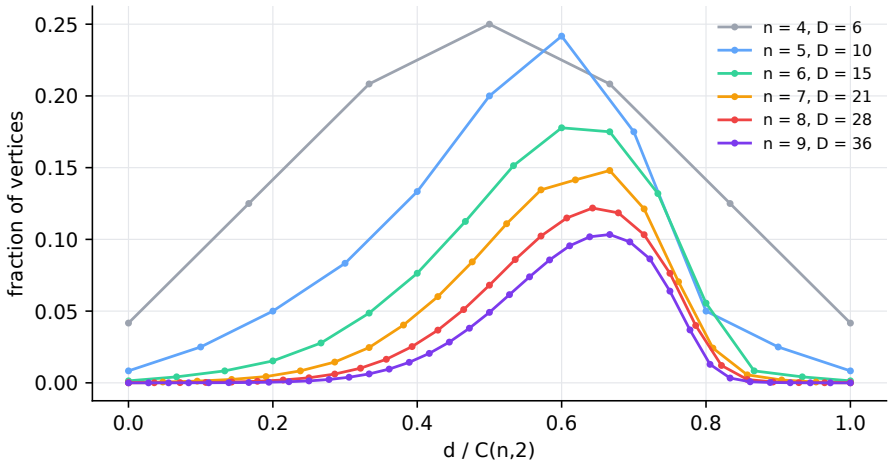


Figure 2: Fraction of vertices at distance d from e, plotted against $d/\binom{n}{2}$. The last point of every curve is the single vertex p^*.

1.3 Organization of the paper

Section 2 describes the overall proof strategy. Section 3 introduces the cursor model and our conventions for permutations and words. Section 4 proves the lower bound; Appendix A supplies the full two-colour and cursor-projection arguments. Section 5 constructs and estimates the words for the upper bound (Figure 4 shows the dependencies between its cases). Section 6 justifies the finite certificate, and Section 7 assembles the cases and explains how to reproduce the computation. Appendices B–D retain proofs of background results, and Appendix E contains

the complete certificate.

1.4 AI statement and agentic system

Key steps of the proof, as well as the Lean formalization, were obtained by our agentic system powered mainly by GPT-6 Astra. For preparation of the text we also used Claude Opus 5.5. The text of the paper is mainly AI-generated, with human edits mostly in the abstract and introduction.

Agentic system. The proof presented in this paper was obtained mainly by using an agentic system developed primarily by one of the authors (S. Uskov), available at the public GitHub repository (link). It is described in more detail in our previous submission that contains the proof for the LX diameter problem [23]. The system is tailored to obtaining rigorous proofs, together with Lean formalizations, with some human orchestration.

2 Strategy of the proof

It is convenient to keep the cards fixed on a circle of n places and to let L and R move a cursor instead (Section 3). A word then consists of *exchanges* (letters X), which change the relative position of the cards, and *rotations* (letters L, R), which only move the cursor. Both bounds come from counting these two kinds of letters separately.

Lower bound. The constant N already appears when only adjacent exchanges are counted. Reversing n cards by exchanges of *adjacent* cards takes exactly $\binom{n}{2}$ exchanges, by the standard equality between adjacent-transposition length and inversion number [24, Proposition 1.5.2]; a proof is recalled in Appendix B.1. In Γ_n an exchange is only available at the cursor, so rotations must be paid for as well. Appendix A shows that a shortest word for p^* separates the labels into two arcs of sizes a, b, $a + b = n$, and exchanges exactly the pairs lying in a common arc. The exchange count is therefore $\binom{a}{2} + \binom{b}{2}$, and when $a, b \geq 2$ projecting the cursor walk to the two arcs gives at least ab rotations. The identity

$$\binom{a}{2} + \binom{b}{2} + ab = \binom{n}{2}$$

explains why the same constant appears in this metric.

The cases where one arc has at most one label are handled by a parity argument; see Theorem 4.1.

Upper bound. Here a word of length at most N has to be produced for *every* permutation. We choose an integer lift d_i of the displacements of the cards, together with the final cursor phase, minimizing the energy $\sum_i d_i^2$. Extended periodically, $F(i) = i + d_i$ is an element of the affine symmetric group, and its inversion-length formula [25] gives the affine length $C(F)$. By Coxeter deletion [24], any LRX word realizing this lift can be reduced to exactly $C(F)$ exchanges without changing its rotations. Hence, with

$$\Delta = \sum_i |d_i|, \qquad D = N - C(F) - \Delta,$$

it suffices to construct a word w with

$$\mathrm{Rot}(w) \leq \Delta + D, \qquad \text{since then} \qquad |w| \leq C(F) + \Delta + D = N.$$

Energy minimization makes d majorized [29] by $(\frac{n-1}{2}, \frac{n-3}{2}, \ldots, -\frac{n-1}{2})$. The displacement Δ is related to Spearman's footrule [26] and to permutation depth [27], and characterizations

of shallow cycles and involutions [28] supply the finite structures used in the contraction and capacity arguments.

The words are then built from explicit cursor routes, with a case analysis according to how the load (the number of arrows $i \mapsto i+d_i$ crossing each edge of the circle) splits into components. All cases are settled by uniform estimates except for one residual branch, which analytic bounds confine to $n \le 2759$ and finitely many parameters; these are cleared by an exhaustive C++ certificate, which also covers $4 \le n \le 10$ by breadth-first search.

Precise statements of the standard results are cited at their points of use. The cursor constructions, route estimates, and bounds for the finite enumeration are proved here.

3 Setup and basic definitions

A cursor instead of shifts. The shift L moves all cards at once, which is hard to follow. Instead, put the cards on a *fixed* circle of n places and mark one place with a cursor. Then

- L and R move the cursor one place forward or back; the cards stay;
- X exchanges the cards at the cursor and at the next place.

Reading the circle from the cursor gives exactly the list of the original problem. In this model rotations are walks of the cursor, and exchanges are the only moves that change the relative position of the cards.

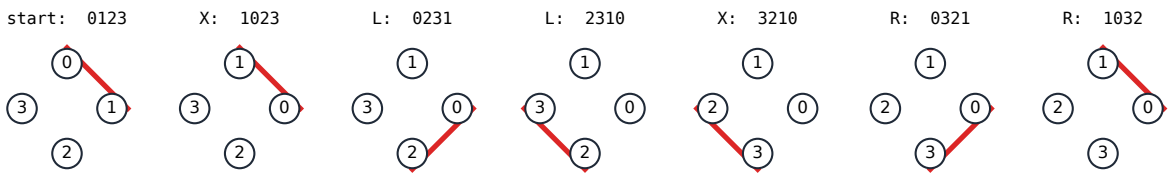


Figure 3: The word $XLLXRR$ in the cursor model ($n = 4$). Red segment: the pair of places (cursor, cursor+1); title: the list read from the cursor. Two exchanges and four cursor steps; the result is $1032 = p^*$.

Permutations and words. The arrangement $(x_0, \dots, x_{n-1})$ is identified with the permutation p, $p(i) = x_i$: it sends a *place* i to the *label* standing there. Label i stands at place $p^{-1}(i)$. A word $w = a_1 a_2 \cdots a_k$ in the letters L, R, X is executed left to right, and $\mathrm{eval}(w)$ is the arrangement obtained from e.

The graph is homogeneous: relabelling the cards is an automorphism, so the graph looks the same from every vertex. Hence it suffices to consider paths from e:

$$\operatorname{diam}\Gamma_n \le B \iff \text{for every } p \text{ there is a word } w \text{ with } \mathrm{eval}(w) = p,\ |w| \le B.$$

4 The lower bound

Theorem 4.1. *For $n \ge 4$ every word taking e to $p^* = (1, 0, n-1, \dots, 2)$ has at least N letters. Hence* $\operatorname{diam}\Gamma_n \ge N$.

Proof. We give a sketch of the proof here, and provide details in Appendix A. Take a shortest word v from e to p^*, with q exchanges and ρ rotations. No pair of labels is exchanged twice: if two letters exchange the same pair, both can be deleted by commuting the relabelling of that pair through the intervening moves.

The target reverses the cyclic order of every triple. Thus each triple contains an odd number of exchanged pairs. The resulting two-colour rule is a special case of the balance theorem [30,31]: exchanged pairs are exactly pairs of the same colour. If the class sizes are a, b, then $a + b = n$ and $q = \binom{a}{2} + \binom{b}{2}$. The colour at every physical place is invariant. Two separated blocks of one colour would contain labels that must be exchanged but can never become adjacent. Hence each nonempty class occupies a single arc.

If a class is empty, $q = N$ suffices. If a class is a singleton, $q = \binom{n-1}{2}$ and $\rho \geq q - 1$, since a shortest word contains no XX. Therefore $|v| \geq (n-1)(n-2) - 1 \geq N$ for $n \geq 5$. For $n = 4$ this gives $|v| \geq 5$; all three generators are odd and p^* is even, so the word length is even and at least $6 = N$.

When $a, b \geq 2$, the cursor projections proved in Appendix A give $\rho \geq \lfloor a^2/2 \rfloor + \lfloor b^2/2 \rfloor$. If both sizes are odd, the endpoint reflection makes ρ odd, improving this bound by one. Consequently $\rho \geq \lfloor (a^2 + b^2)/2 \rfloor \geq ab$, and

$$|v| = q + \rho \geq \binom{a}{2} + \binom{b}{2} + ab = N.$$

Appendix A proves the exchange, colour, reflection, and projection claims in full, including all endpoint cases. □

A word of length exactly N reaching p^* exists (for $n = 4, 5, 6$: $XLLXRR$, $XLLXLXRXRR$, $XLXRXLLLXLXRXRR$), so $d(e, p^*) = N$; this also follows from the upper bound.

5 The upper bound: a complete constructive proof

For $4 \le n \le 10$ we use the exhaustive BFS in Section 6. Throughout the construction below, $n \ge 11$. The complete finite certificate is in Appendix E. Here p denotes the parity of n, q counts nontrivial cycles, and p_i is the position of token i in the target list. Put

Notation	Definition or role
m, p, h	$n = 2m + p$, $p \in \{0, 1\}$, $h = \lfloor (n-1)/2 \rfloor$.
Q, N	$Q = \lfloor n^2/4 \rfloor = m(m+p)$, $N = 2Q - m = \binom{n}{2}$.
F, d_i	$F(i) = i + d_i$ is the balanced periodic destination map.
Δ, C	$\Delta = \sum_i \lvert d_i \rvert$; C is affine length (5.2).
f, q, rk	$f = \#\{i : d_i = 0\}$, q counts nontrivial cycles; $\mathrm{rk} = n - f - q$.
$Z, \mathcal{L}$	$Z = \Delta - C - \mathrm{rk}$, $\mathcal{L} = \sum_B (\lvert B \rvert - 2)$.
ε, δ, A	$\varepsilon = Q - \Delta$, $\delta = Q - m - C$, $A = 2\varepsilon + p - f$.
D	$D = N - C - \Delta = \delta + \varepsilon$, the available rotation excess.

Packet and gap parameters $(r, k, b_i, w_i, U, S, K)$ are defined where used. The head coordinate H in Section 5.3 is local to that construction; later H denotes a displacement deficit. It suffices to construct a word with at most $\Delta + D$ rotations: after deleting redundant exchanges, its length is at most $C + \Delta + D = N$. We call an estimate paid when its excess $\mathrm{Rot} - \Delta$ is at most D. Every route below allows arbitrary initial and final physical cursors. Only the cursor of each individual service procedure is closed. Connected load is settled in Section 5.5. A zero edge permits a finite cut in Section 5.6. We then settle small invariant blocks and a large shallow packet before separating the two-component case (Section 5.9) from the remaining case (Sections 5.10–6).

5.1 Choosing a lift and its affine invariants

5.1.1 Minimization over the final phase

If token i has position p_i in the target list, its final physical position at cursor c is $p_i + c \pmod n$. Choose integers d_i with

$$\sum_i d_i = 0, \qquad i + d_i \equiv p_i + c \pmod n. \tag{5.1}$$

Such a lift exists: take $c = 0$ and $d_i = p_i - i$. Minimize the nonnegative integer energy $\mathcal{E}(d) = \sum_i d_i^2$ over all lifts and phases. Extend d_i periodically and put $F(i) = i + d_i$ on $\mathbb{Z}$. Distinct residues of $F(i)$ give a bijection with $F(i+n) = F(i) + n$; balance places it in the affine symmetric group. Its standard length is [25, Proposition 3.1]

$$C(F) = \sum_{0 \le i < j < n} \left| \left\lfloor \frac{F(j) - F(i)}{n} \right\rfloor \right|. \tag{5.2}$$

This counts affine inversion orbits and is the minimum number of periodic adjacent exchanges. Appendix B.2 checks our indexing convention.

Lemma 5.1. *If $P_j(d)$ is the sum of the j largest coordinates, the chosen lift satisfies*

$$P_j(d) \le \frac{j(n-j)}{2} \quad (1 \le j < n). \tag{5.3}$$

In particular, $\lvert d_i \rvert \le h < n/2$.

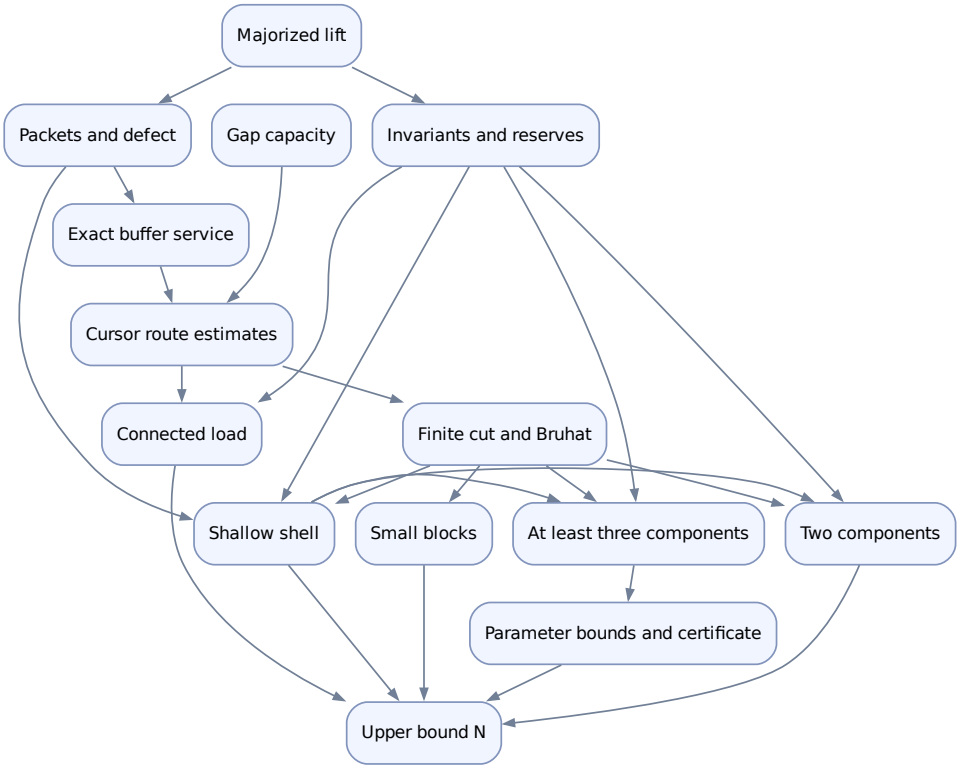


Figure 4: Dependencies in the upper proof. The connected and finite-load branches cover all lifts; within the latter, the final case uses the bounded certificate.

Proof. For a set S of j indices put $d'_i = d_i + j - n\mathbf{1}_{\{i\in S\}}$. The sum remains zero and the phase becomes $c + j$. Expanding squares gives

$$\mathcal{E}(d') - \mathcal{E}(d) = n\Big(j(n-j) - 2\sum_{i\in S} d_i\Big) \geq 0.$$

Choose S with largest sum. The cases $j = 1, n-1$, together with balance, give strict shortness. □

Condition (5.3) is ordinary majorization $d \prec ((n-1)/2, (n-3)/2, \ldots, -(n-1)/2)$ [29, Section 3, Theorem 7]. Appendix C retains the notation and the full energy expansion.

We call a lift satisfying (5.3) *original*. Auxiliary maps need not satisfy this condition. The chosen arrow of i is the directed arc of length $|d_i|$ from i to $i + d_i$. The load of an edge counts arrows crossing it in either direction. The winding of a cycle B is $w_B = n^{-1}\sum_{i\in B} d_i \in \mathbb{Z}$. A position is active if $d_i \neq 0$. A port is a cursor position a, identified with the edge between a and $a + 1$; the ports incident to vertex v are $v - 1, v$. For a cluster $\mathcal{B}$, its positive service ports are

$$\mathcal{E}_{\mathcal{B}} = \{a : \{a, a+1\} \cap \operatorname{supp} \mathcal{B} \neq \varnothing,\ \lambda_{\mathcal{B}}(a) > 0\}.$$

A positive packet port below always means a port in $\mathcal{E}_{\mathcal{P}}$. This incidence condition is needed to apply the service lemma. Load components are the components of positive edges; isolated unloaded vertices are not counted. For a balanced cluster, its signed edge flux (the number of clockwise crossings minus counterclockwise crossings) is constant: at each vertex one incoming

and one outgoing arrow cancel. Summing over the n edges shows that this constant is $\sum_i d_i/n = 0$. Thus its load $\lambda(a)$ is even on every edge. Also

$$\Delta = \sum_{a=0}^{n-1} \lambda(a),$$

since an arrow of length $|d_i|$ contributes to exactly $|d_i|$ edges. A half-load means $\lambda(a)/2$; a half-peak is its maximum. A lift is finite if it preserves some window of n consecutive integer positions; its inversion number is then the ordinary one in that window.

5.1.2 Deleting redundant exchanges

Periodic adjacent transpositions form a Coxeter system [25, Proposition 3.3]. By the deletion property [24, Proposition 1.4.7], pairs of X letters can be deleted from any word with lift F until exactly $C(F)$ exchanges remain. Every L, R letter and the lifted endpoint are preserved: the location of a remaining exchange depends only on the rotations already traversed. Appendix B.2 gives the application and its trajectory interpretation.

5.1.3 Sorting across fixed intervals and the defect

For a finite permutation π on b positions put

$$I(\pi) = \#\{(i,j) : 0 \le i < j < b,\ \pi(i) > \pi(j)\}.$$

Its reflection length is $\mathrm{rk}(\pi) = b - \#\{\text{all cycles of } \pi\}$, the standard length in arbitrary transpositions [26, Section 2]; its proof is retained in Appendix B.1. For the residue permutation of a lift this is $n - f - q$. If the lift preserves a common finite window, (5.2) is the ordinary inversion number in that window.

Choose consecutive active positions u, v whose residual displacements are $a > 0$ and $-b < 0$; all intermediate positions are fixed. If their chosen arc has length ℓ, then $a, b \ge \ell$: a shorter displacement would have a destination already occupied by a fixed intermediate token. The palindrome of adjacent exchanges on edges $u, \dots, v-1, v-2, \dots, u$ exchanges the endpoints and restores the intermediate tokens. The new residuals are $\ell - b, a - \ell$. Every exchange is a descent: the destinations of the moving tokens lie on opposite sides of the fixed intermediate destinations. Hence

$$\Delta_{\mathrm{old}} - \Delta_{\mathrm{new}} = 2\ell, \qquad C_{\mathrm{old}} - C_{\mathrm{new}} = 2\ell - 1. \tag{5.4}$$

Active positions only disappear; a token keeps its sign until its residual vanishes, and $\ell \le h$. Balance supplies another such pair until the identity is reached. If there are M endpoint transpositions, then

$$M = \Delta - C, \quad M \ge n - f - q, \quad M \equiv n - f - q \pmod 2. \tag{5.5}$$

The last two assertions are the usual reflection length and parity of a product of transpositions. Thus $Z \ge 0$ and Z is even. On the finite line the same proof needs no half-circle restriction. There Δ is Spearman's footrule (total displacement), and $\Delta = 2\,\mathrm{dp}$ identifies it with twice the depth [27, Theorem 1.1, Remark 1.2]. The inequality $I + \mathrm{rk} \le \Delta$ is the Diaconis–Graham inequality [26, Theorem 2]. We have proved the affine version needed here.

Lemma 5.2. *For an original lift,*

$$\varepsilon \ge 0, \quad 4\varepsilon \ge f^2 - p, \quad A \ge 0, \quad D = m - q + Z + A, \quad 2\delta = \mathcal{L} + 2Z + A, \tag{5.6}$$

where $\mathcal{L} = \sum_B (|B| - 2)$ over nontrivial cycles. Also Δ is even, $\varepsilon \equiv Q \pmod 2$, and $\varepsilon \ge f - 1$.

Proof. If the sign counts are a, b, with $a + b = n - f$, (5.3) gives $\Delta \le \min(a(n-a), b(n-b)) \le (n^2 - f^2)/4$. This proves the first two bounds. For $f \ge 2$, $2\varepsilon \ge (f^2 - p)/2 \ge f - p$; for $f = 0$, $A = 2\varepsilon + p \ge 0$; for $f = 1, p = 0$, integrality gives $\varepsilon \ge 1$. Thus $A \ge 0$. Substitute $\mathcal{L} = n - f - 2q$ to get the identities. Balance makes Δ twice the positive sum. Finally, $f^2 - p \ge 4(f-1)$ for $f \ge 3$; the cases $f = 0, 1, 2$ follow directly using integrality. □

5.2 Original root packets

For a nontrivial cycle B, the union E_B of its arrows is a connected proper arc or the whole circle. For distinct nontrivial cycles put $B \to C$ when E_B contains a vertex of C. That vertex is strictly inside an arrow of B, since the cycle supports are disjoint. These arrows define the passage graph on nontrivial cycles. A cycle reaches itself by the empty path. A source strongly connected component has no incoming edge from another strongly connected component; we call it a root packet when it is the source in Lemma 5.3.

Lemma 5.3. *Each positive-load component has a unique source strongly connected component $\mathcal{P}$ in the passage graph. Every cycle in it reaches all other cycles of the load component. Cycles outside $\mathcal{P}$ lie in invariant open gaps of its support. If the original load is connected, the packet is balanced.*

Proof. If E_B meets E_C without being contained in it, then $B \to C$: inspect the endpoints of proper arcs; a full circle contains all vertices. Proper arcs have distinct endpoints because cycle supports are disjoint. Choose an inclusion-maximal arc. If it does not reach everything, some intersecting pair E_B, E_C has B reached and C unreached. Necessarily $E_B \subset E_C$. Along a path from the chosen maximal arc to B, the first entry into an arc contained in E_C gives a passage from the preceding cycle to C, a contradiction. Thus some cycle reaches all, and the source strongly connected component is unique.

An outside cycle cannot pass through a packet vertex, since that would give reverse reachability. Its arrows and vertices therefore lie in a single open gap of the packet support; its winding is zero. With connected load, all nonzero windings belong to the packet: such a cycle covers the circle and passes through every other cycle. The windings of the whole lift sum to zero, so the packet is balanced. □

Let the packet have r active vertices and k cycles. The quantities $\Delta_{\mathcal{P}}, C_{\mathcal{P}}, Z_{\mathcal{P}}$ refer to the map retaining the packet and fixing its complement. Its internal gaps have sizes b_i and constant even packet loads $w_i \ge 2$. When the entire positive load is connected and the packet does not cover the circle, its external gap contains only fixed positions: an active cycle there would form a separate load component. Denote their number by e; otherwise put $e = 0$. An arrow crossing an internal gap has length at least $b_i + 1$, so

$$b_i \le h - 1 = m + p - 2. \tag{5.7}$$

Lemma 5.4. *For connected original load,*

$$Z_{\mathcal{P}} \ge k - 1, \quad Z \ge Z_{\mathcal{P}}, \quad \delta \ge \lceil r/2 \rceil + Z_{\mathcal{P}} - k \ge \lceil r/2 \rceil - 1. \tag{5.8}$$

For a finite packet, the stronger bound $Z_{\mathcal{P}} \ge 2(k-1)$ holds.

Proof. Apply (5.4) to the packet alone. Its event graph has the initial active packet positions as vertices and one undirected edge $\{u, v\}$ for each endpoint transposition; repeated events are counted with multiplicity. The chords of this graph do not cross: no active vertex is inside the arc of an event, and the endpoints of later events are already active at that time. Different graph components form a noncrossing circular partition. Each original cycle lies in a graph component: the product of the endpoint transpositions preserves every graph component. Moreover, the

residual displacement of a token never changes sign in (5.4). Its successive event arcs therefore cover its original arrow monotonically.

If an original arrow from one graph component passes a vertex of another, noncrossing places the whole second component in an open gap between consecutive vertices of the first. The monotone event path covering that arrow contains an event crossing the entire gap; otherwise it would have an endpoint inside the gap belonging to the first component. That event has length at most h, so the gap has width at most $h < n/2$. The shortest arrows of the second component stay inside it, since their endpoints lie in this arc. By noncrossing, subsequently reached components also stay there, so return to the first component is impossible. On the line the same assertion follows from nested intervals without the bound h. Thus no directed path through distinct graph components can return to the first one. A strongly connected component of cycles lies in one event graph component. For a packet the event graph is connected, hence $M_{\mathcal{P}} \geq r - 1$. Equation (5.5) gives $Z_{\mathcal{P}} = M_{\mathcal{P}} - r + k \geq k - 1$.

Inversions are additive between the packet and its invariant gaps: outside destinations are outside each gap and all its periodic copies, so permuting inside destinations preserves every outside comparison. Displacement and reflection length are additive too. Thus $Z = Z_{\mathcal{P}} + \sum_i Z_i \geq Z_{\mathcal{P}}$. Using $\mathcal{L} \geq r - 2k$ in (5.6) proves the bound on δ.

For the stronger finite bound, read the events backwards. Each edge that first connects two graph components merges two cycles of the current product. There are at least $r - 1$ such merges. If M_+, M_- count merges and splits, $M_+ - M_- = r - k$, whence $M_- \geq k - 1$ and $Z_{\mathcal{P}} = M_+ + M_- - r + k = 2M_- \geq 2(k - 1)$. □

5.3 Exact words using a shared logical buffer

Lemma 5.5. *Choose a nontrivial root cycle and a vertex v on it. Suppose that a balanced cluster contains the root cycle and every one of its cycles is reachable from it in the passage graph induced by the cluster, and every chosen arrow has length less than n. At either port $a = v - 1, v$ there is an exact word whose integer cursor closes, with rotation count*

$$\mathrm{Rot} = \Delta_{\mathrm{cluster}} - \Lambda_{\mathrm{cluster}}(a). \tag{5.9}$$

Here $\Lambda_{\mathrm{cluster}}(a)$ is the total load of the cluster's chosen arrows at edge a. After redundant exchanges are deleted, C letters X remain.

Proof. For a right port translate the root vertex to 0. Put $M = n - 1$, $F_0(x) = x - \lceil x/n \rceil$, and $\tau_i = F_0(i + d_i) - F_0(i)$. Token 0 goes into a logical buffer; the other tokens occupy classes $y_i = F_0(i)$ modulo M. The head has integer coordinate $H = 0$. The lifted physical positions of a passive token and the buffer token are, respectively,

$$P_H(y) = y + 1 - \left\lceil \frac{H - y}{M} \right\rceil, \qquad B_H(y) = H + n\frac{y - H}{M}. \tag{5.10}$$

For the buffer, $y \equiv H \pmod{M}$. At $H = 0$ these are the initial positions.

The macro $U = XL$ increases both H and the buffer coordinate by one; $V = RX$ decreases both by one; X exchanges the buffer and the token in class $H \pmod{M}$, without changing their y coordinates. Substitution in (5.10) checks all three operations, including crossings of multiples of n.

Follow each cycle in destination order. Move each buffer token by τ_i using U, V, then transfer the buffer to the next token by X. The root procedure is open: omit the initial and final X. For a child cycle, include an initial X and a final X restoring the incoming buffer. Without recursive insertions, a child of winding w adds exactly τ_i to its tokens, leaves all other y coordinates unchanged, and shifts H by Mw. Indeed $\sum_{i \in B} \tau_i = Mw_B$ follows by telescoping around the cycle and using $F_0(x+n) = F_0(x) + M$. With insertions, the same statements hold for the entire served subtree, with w replaced by the sum of its cycle windings. This follows by induction on

the number of inserted cycles; each insertion restores the incoming logical buffer and changes no other coordinates outside its own subtree.

Before each macro, recursively start an unstarted cluster cycle whose passive token is at the head, if there is one. Mark it as started before recursing, so the finite recursion terminates. An arrow passing strictly through a passive vertex guarantees an encounter: F_0 preserves their order and the head visits every intermediate integer coordinate. The only collapsed pair is $0, 1$ modulo n. A negatively directed arrow ending at 0 could encounter vertex 1 only at its final head class. That class is already checked before the first root macro. If the root has no macro before its first transfer, then $\tau_0 = 0$, hence $d_0 = 1$, and vertex 1 belongs to the root cycle itself. Thus this exception cannot leave a different cycle unstarted. An insertion shifts H by a multiple of M, so the parent's remaining head classes are preserved. If a reachable cycle remained unstarted, the first unstarted cycle on a directed path from the root would have been encountered by its served predecessor. Hence every cluster cycle is served.

At the end $y_i = F_0(i + d_i)$ and $H = M \sum w = 0$. The final root token has target residue 0 and remains in the buffer. Equation (5.10) gives exactly $i + d_i$, including winding. The number of U, V macros is $\sum_i |\tau_i|$. The contraction F_0 removes one step precisely from arrows crossing the chosen edge; each arrow crosses at most one copy of that edge. This proves (5.9). Reflection gives the left-port version. □

Example of the service. Let $F(0) = 1$, $F(1) = 2$, $F(2) = 0$, fixing all other positions, and choose root vertex 0 and port 0. Then $(\tau_0, \tau_1, \tau_2) = (0, 1, -1)$, $\Delta = 4$ and $\Lambda(0) = 2$. The procedure gives $XUXV = XXLXRX$. Deleting the initial XX leaves $LXRX$, which sends tokens $0, 1, 2$ to positions $1, 2, 0$ and restores cursor 0. Its two rotations equal $4 - 2$, and its two exchanges equal $C = 2$.

Example with opposite windings. Let $n = 12$, with $d_i = 2$ for even i and $d_i = -2$ for odd i. The cycles $(0, 2, 4, 6, 8, 10)$ and $(1, 11, 9, 7, 5, 3)$ have windings $1, -1$. Take root vertex and port 0. The odd cycle is inserted at the first head encounter. It restores logical buffer token 0 with $y_0 = 0$, but changes H from 0 to -11. Its physical position is then $B_{-11}(0) = 1$, so this child procedure does not close physically. The even cycle contributes the remaining 11 head steps and the full service ends at $H = 0$, with every token at $i + d_i$. Here $\Delta = 24$, $\Lambda(0) = 2$, $C = 12$. After deletion one possible word is

$$X(RRX)^5RX(LLX)^5L,$$

with 22 rotations and 12 exchanges. Figure 5 shows the head positions during this service.

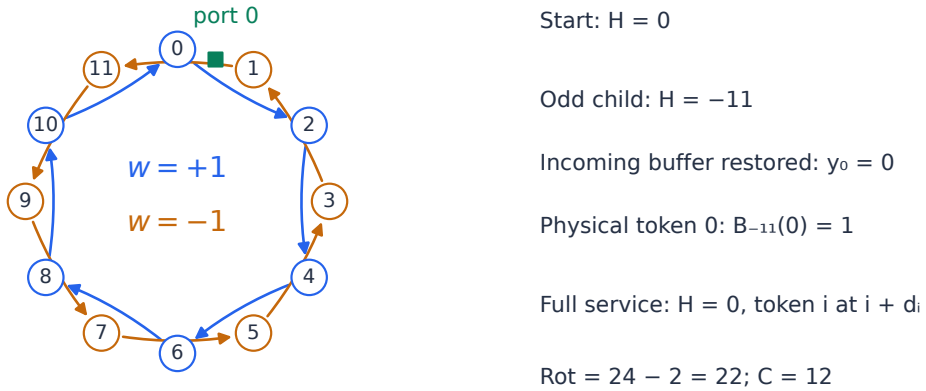


Figure 5: The $n = 12$ service with two opposite windings. A child restores the logical buffer before the balanced full service closes the physical cursor.

A child procedure with nonzero winding need not close physically on its own. The proof maintains a shared logical buffer. Only completed balanced maps with disjoint supports may be reordered. In particular, finite filling cycles can be served separately on an outside tour. The packet and all cycles left over are served together: an arrow of some packet cycle passes directly through each remaining cycle.

From a walk to a word. For a balanced cluster whose root packet reaches every cycle, choose a physical walk γ from the initial cursor to the final one, visiting a packet port. At a visited incident port, one may serve a finite filling cycle separately; the root service then contains the packet and every filling not served separately. These clusters are disjoint and balanced, and each service restores its cursor. Therefore insertions preserve the walk and give

$$\mathrm{Rot} \le \Delta + |\gamma| - \lambda_{\mathcal{P}}(a_{\mathcal{P}}) - \sum_{B \in \mathcal{S}} \lambda_B(a_B),$$

where $\mathcal{S}$ is the set of separately served cycles and all selected ports lie on γ. The root may be any packet cycle incident to its selected port, since every packet cycle reaches the whole component. Each service realizes its own destination map and fixes the other tokens, so composing the services gives the required map. This is the pricing rule used in all route estimates below.

5.4 A quadratic estimate for a filling

For finite permutations, the classical bound is $\Delta \le 2\lfloor b^2/4 \rfloor$ [27, Proposition 3.2]. We need the following refinement with a prescribed fixed-point count.

Lemma 5.6 (Finite capacity)**.** *If a permutation π on b positions has f fixed points and $l_0 = \lfloor (b-f)/2 \rfloor$, then*

$$\Delta(\pi) \le 2l_0(b - l_0) \le (b^2 - f^2)/2.$$

Proof. Let the smaller sign count be $z \le l_0$. The sum of its arrow lengths is at most the sum of the z largest positions minus the sum of the z smallest, namely $z(b-z)$. Balance doubles this sum, and $z(b-z)$ is increasing on $[0, b/2]$. □

For a filling, locally number its vertices $1, \dots, b$ and its ports $0, \dots, b$: port j is the edge between j and $j+1$. Ports $0, b$ are incident to the two packet boundary vertices outside the filling, and the filling itself has zero load at those ports. Let π be the filling permutation. For each nontrivial cycle B, let λ_B be its load and $\mathcal{E}_B$ its positive incident ports. Define

$$\begin{aligned} L(y) &= \sum_B \max\big(\{\lambda_B(a) : a \in \mathcal{E}_B, a \le y\} \cup \{0\}\big), \\ R(x) &= \sum_B \max\big(\{\lambda_B(a) : a \in \mathcal{E}_B, a \ge x\} \cup \{0\}\big), \\ K(\pi) &= \max_{0 \le x \le y \le b} \min\{x + y - L(y),\ 2b - x - y - R(x)\}, \end{aligned}$$

where x, y are integers. The two visited parts cover all filling ports. If the filling has q_0 nontrivial cycles, then

$$0 \le K(\pi) \le b - q_0. \tag{5.11}$$

For the upper bound average the two expressions: the sum of the two available maxima of each cycle is at least two. For the lower bound take $x = y = 0$: $L(0) = 0$ and the sum of individual cycle peaks is at most b.

Lemma 5.7. *Every finite filling satisfies*

$$2\Delta(\pi) + K(\pi)^2 \le b^2. \tag{5.12}$$

Proof. Let a_j be half the total load at cut j. Then $a_0 = a_b = 0$, $a_j \geq 0$, $|a_{j+1} - a_j| \leq 1$, and $\Delta = 2\sum_{j=1}^{b-1} a_j$. Its linear interpolation $a(t)$ is nonnegative, 1-Lipschitz, and has integral $\Delta/2$. A cycle's load changes only at its incident ports. Hence $2a(t) \leq L(y)$ for $t \leq y$, and $2a(t) \leq R(x)$ for $t \geq x$.

Choose a pair attaining K and reflect so that $x + y \leq b$. Put $k = K/2$, $u = (x + y - K)/2$, $v = (2b - x - y - K)/2$. Then $0 \leq u \leq v$, $b = u + v + 2k$, and $z = (x + y)/2 = u + k$. The zero endpoints and the Lipschitz bound give

$$a(t) \leq \min(t, u) \quad (0 \leq t \leq z), \qquad a(z + t) \leq \min(u + t, v, v + k - t) \quad (0 \leq t \leq v + k).$$

Writing $A = v - u$, the sum of these two upper integrals is

$$\begin{cases} uv + k(u + v), & A \leq k, \\ (u + v)^2/4 + k(u + v) - k(2A - k)/4, & A \geq k. \end{cases}$$

Indeed the left integral is $u^2/2 + uk$. The constraint $u + t$ subtracts $A^2/2$ from the right integral $v^2/2 + vk$ if $A \leq k$, and $A^2/4 + Ak/2 - k^2/4$ if $A \geq k$. Both expressions are at most $(u + v)^2/4 + k(u + v) = (b^2 - K^2)/4$, proving (5.12). The case $b = 0$ is immediate. □

5.5 The entire connected case

Assume the positive load of the original lift is connected. Write $r = 2l + \eta$, $\eta \in \{0, 1\}$ for the packet support size, k for its cycle count, and $s = n - r = \sum b_i + e$.

5.5.1 Full tours and the external arc

If both complementary arcs between the two cursors meet positive packet ports, their lengths sum to n. The sum of the two root discounts is at least four, and each outside cycle contributes at least two. Taking the shorter complete tour gives

$$\mathrm{Rot} - \Delta \leq m - (q - k) - 2 \leq D, \tag{5.13}$$

where the last inequality uses $Z \geq k - 1$ and $A \geq 0$.

Otherwise the cursors lie in one gap between ports. For an internal gap, the walks $x \to 0 \to y$ and $x \to b \to y$, serving visited filling cycles separately, give

$$\mathrm{Rot} - \Delta \leq K_i - w_i, \qquad w_i \geq 2. \tag{5.14}$$

For an external gap of e fixed positions, the boundary positive ports are $e + 2$ edges apart and have load two. The excess is at most e.

We pay for it by $D \geq f \geq e$. To prove this in the connected case, use (5.6) to obtain

$$D - f \geq \mathcal{L}/2 + Z + f(f - 3)/2.$$

This suffices for $f = 0$ or $f \geq 3$. For $f = 1, 2$, integrality leaves only $\mathcal{L} = Z = 0$. All cycles are then transpositions, whose chords do not alternate: the event graph has minimum size and its components are exactly those pairs. Connected load therefore has an outer pair enclosing all $n - f$ active vertices, of length at least $n - f - 1$. For $n \geq 5$ and even $n - f$ this exceeds h. The remaining case $n = 4$, $f = 2$ is one adjacent pair, with $C = 1$, $\Delta = 2$, $D = 3 \geq f$.

5.5.2 Joint displacement and the complement of a gap

Put

$$G = \left\lfloor (s+\eta)^2/4 \right\rfloor, \quad H = l(n-l) - \Delta_{\mathcal{P}}, \quad S_2 = \sum_i b_i^2, \quad T = S_2 - 2G - 2H.$$

We have $H \geq 0$: averaging the coordinates of each discarded zero-winding cycle to zero preserves (5.3), since P_j is convex. The remaining balanced core has smaller sign count at most l, hence $\Delta_{\mathcal{P}} \leq l(n-l)$. Also $Q - G = l(n-l)$. Apply Lemma 5.7 to one filling and $2\Delta_j \leq b_j^2$ to the others to get

$$K_i^2 \leq T + 2\varepsilon. \tag{5.15}$$

If every $b_i \leq s/2$, then $S_2 \leq s^2/2$ and $2T \leq 1$. Otherwise there is a unique $b > s/2$. Let c be the sum of the other internal lengths, $t = b - c - e = 2b - s > 0$, and $u = t - \eta$. Using $S_2 \leq b^2 + c^2$ and $4G = (s+\eta)^2 - p$ gives

$$2T \leq u^2 + p - 2\eta - 4H. \tag{5.16}$$

The discarded terms are $-4\eta(c+e) - 4ce - 2e^2 \leq 0$.

The core occupies $n - b$ consecutive positions in the complement of the gap. Represent its residue permutation in this finite interval, retaining the physical distances. Original arrows are shortest, so their lengths are at most the corresponding finite lengths. A permutation on L positions with $r = 2l + \eta$ active vertices has displacement at most $2l(L-l)$: the smaller sign count z is at most l, its length sum is at most $z(L-z)$, and this increases up to $L/2$. Consequently

$$\Delta_{\mathcal{P}} \leq 2l(n-b-l), \qquad H \geq l(u-l). \tag{5.17}$$

No original majorization is assumed for this finite map. The inequalities $b \leq n/2$ and $t \geq 1$ give $0 \leq u \leq 2l$. Together with $H \geq 0$, (5.16)–(5.17) imply

$$2T \leq \min\{u^2, (2l-u)^2\} + p - 2\eta \leq l^2 + 1. \tag{5.18}$$

This bound also holds in the nondominant case.

If (5.14) is not paid, then $K_i - w_i > \delta + \varepsilon$. Equation (5.8) and integrality give $K_i \geq l + \varepsilon + 2$. But (5.15) then implies

$$T \geq (l+\varepsilon+2)^2 - 2\varepsilon = (l+2)^2 + \varepsilon(2l+\varepsilon+2) > (l^2+1)/2,$$

a contradiction. Every internal gap is paid. Together with the external case and (5.13), this proves the following.

Theorem 5.8. *Every original lift with connected positive load admits an exact LRX word of length at most N for any two physical cursors. A common finite cut and zero individual windings are not required.*

5.6 Joining components and transferring a word

Suppose positive load has $\rho \geq 2$ components. No cycle has nonzero winding, since it would cover the circle. Cut at a zero edge to get finite invariant intervals. The size of a component counts positions from its first through its last vertex, including any internal fixed positions. Call a component small if its size is at most $\lceil n/2 \rceil$ and large otherwise; at most one component can be large. Choose a packet in each component by Lemma 5.3. If their union has r vertices and k cycles, then $Z_{\mathcal{P}} \geq k - \rho$, $\delta \geq r/2 - \rho$, and $r \geq 2\rho$.

5.6.1 Bridges between components

Let $g_1, \dots, g_\rho \geq 1$ be the zero separator widths in edges; their internal vertices are fixed. Leave a widest separator g_* as the window cut and join across the others. Put $B = \sum_j g_j - g_* \leq f + \rho - 1$. For one join let u be the rightmost vertex of the left component and v the leftmost vertex of the next, with $g = v - u$. Their incoming arrows have origins ordered $a < u < v < b$. Exchange target values u, v. Both lengths increase by g. There are $g - 1$ fixed positions between u and v, so inversions increase by $2g - 1$. Two cycles merge, and load increases by two exactly on the bridge.

All old passages persist because the changed arrows only lengthen. The cycles containing extreme vertices belong to their packets: an outside cycle lies strictly in a packet gap and cannot contain a component endpoint. After all joins the original packets form one reachable packet. The resulting map G satisfies

$$\Delta(G) = \Delta + 2B, \qquad C(G) = C + 2B - \rho + 1. \tag{5.19}$$

Its arrows lie in one finite window and have length less than n. We do not use (5.3) for G.

5.6.2 Deleting comparisons by rank order

For a finite permutation q put $\mathcal{R}_q(i,t) = \#\{j \leq i : q(j) > t\}$ for $0 \leq i, t < n$, and $\mathcal{R}_q(-1,t) = 0$. Write $p \preceq q$ if every rank number for p is at most the corresponding one for q. This is the strong Bruhat order [24, Theorem 2.1.5]. An increasing target exchange raises rank numbers on a rectangle and preserves the others; hence $F \preceq G$. Reversing an invariant interval also gives an upper map: repeatedly exchange adjacent increasing values until they are in decreasing order. Each exchange raises the rank numbers.

Lemma 5.9. *Every reduced adjacent sorting word for G sorts F if its comparisons are performed and already ordered pairs are skipped. Exactly $C(F)$ exchanges remain. All cursor movements and both cursor endpoints are preserved.*

This follows from the lifting property of Bruhat order [24, Proposition 2.2.7], with the rank criterion [24, Theorem 2.1.5]. Appendix D retains the inductive rank proof.

A destination map G corresponds to the physical list G^{-1}. Inversion preserves Bruhat order [24, Corollary 2.2.5], so $F^{-1} \preceq G^{-1}$. Apply the comparison statement to these two lists in the fixed window. For a word generating the map, first read it backwards as a sorting word, delete comparisons, and reverse again. Thus the original cursor endpoints are preserved. Every service exchange for a finite map lies inside the window: all child windings in Lemma 5.5 are zero, H is restored, and the head stays in the contracted interval of the arrows. If the extreme physical vertices are a, z, the service cursor stays between a and $z - 1$. Thus transfer preserves the exact integer lift.

5.6.3 The general route estimates

Let $2U$ be the largest individual peak of an original packet. For each filling cycle C, write its peak as $2U_C$ and put $S = \sum_{C \text{ outside packets}} U_C$. The joined packet has peak $2U$: new bridges have load two and old interval loads are unchanged. For complementary arcs both meeting the packet, the sum of available root peaks is at least $2U + 2$; for each filling it is at least $2U_C$. Using (5.19), service, and transfer gives

$$\mathrm{Rot} - \Delta \leq 2B + m - U - S - 1. \tag{5.20}$$

For the other cursor positions, the excess bounds are

$$2B + K_i - w_i, \qquad 2B + g_j - 3, \qquad 2B + g_* - 1 \tag{5.21}$$

for an old internal gap, a new bridge, and the external arc, respectively. A bridge has $g_j - 1$ fixed internal positions, so $K = g_j - 1$, while its root load is two. The external arc has $g_* - 1$ fixed positions. These observations account for every constant in the formulas.

Define the unpaid excess of the two-arc estimate by

$$P = 2B + m - U - S - 1 - \delta - \varepsilon. \tag{5.22}$$

If q_{out} counts filling cycles, then $S \geq q_{\mathrm{out}}$, $Z \geq k - \rho$, and (5.6) gives

$$P \leq 2B + \rho - U - 1 - A. \tag{5.23}$$

5.7 Finite inversion and structural lemmas

5.7.1 Deficit from reversal

Lemma 5.10. *For a permutation π on b positions, with $\gamma = \binom{b}{2} - I(\pi)$,*

$$K(K-1) \leq 2\gamma, \qquad K \leq \gamma + 1.$$

Proof. Let S_0 be the sum of individual half-peaks. Then $2S_0 \leq b - f$. The choice $x = 0, y = b$ gives $K \geq b - 2S_0 \geq f$. By (5.5), $\Delta - I \geq \mathrm{rk}(\pi) \geq (b-f)/2 \geq (b-K)/2$. Together with (5.12), this yields $2I \leq b^2 - K^2 - b + K$, the first inequality. For $\gamma = 0$ it gives $K \leq 1$; for $\gamma \geq 1$, a value $K \geq \gamma + 2$ would contradict it. □

Lemma 5.11. *A balanced strictly short lift T satisfies $C(T) \leq Q - m$. If T retains the original packets and reverses their internal gaps, and, when needed, the entire external interval of a large component, then a component of size greater than $\lceil n/2 \rceil$ implies $C(T) \leq Q - m - 1$.*

Proof. Put $g(i) = T(i) + h$ and $\sigma(i) = g(i) \bmod n \in [0, n)$. Since $i \leq g(i) \leq i + 2h < i + n$, we have $g(i) = \sigma(i) + n$ exactly when $\sigma(i) < i$. There are h such positions, since $\sum_i (g(i) - i) = hn$. Put $\mathrm{dep} = \sum_{\sigma(i)<i} (i - \sigma(i))$. Splitting pairs according to this set and its complement gives

$$C(T) = h(n-h) + I(\sigma) - 2\,\mathrm{dep}\,. \tag{5.24}$$

Indeed there are no inversions with left position in the set and right position outside it. The number of inversions of the opposite type is dep: summing the difference between left and right inversion counts, which is $i - \sigma(i)$, cancels internal pairs. For each cross pair the affine contribution complements the finite one to one, proving (5.24). By (5.5), $2\,\mathrm{dep} - I(\sigma) \geq \mathrm{rk}(\sigma)$. A cycle has at most its size minus one positions with $\sigma(i) < i$, so $\mathrm{rk}(\sigma) \geq h$. Thus $C(T) \leq h(n-h-1) = Q - m$.

For strictness, every new arrow has length less than h: internal gaps have size at most $h-1$, and the external interval at most $m-1$. Coordinates $\pm h$ are therefore inherited from the original lift. By (5.3), there are at most $n-2h$ coordinates $-h$; for odd n, there is at most one coordinate $+h$. Equality $C(T) = Q - m$ would force $\mathrm{rk}(\sigma) = h$ and $Z(\sigma) = 0$. There are at least $n - 2h$ fixed points of σ, each corresponding to a coordinate $-h$. Hence there are exactly $n - 2h$ fixed points, and all other cycles are pairs.

Since σ is a shallow involution, its matching chords do not cross [28, Corollary 25]. A direct four-endpoint proof is given in Appendix D.2. The following restrictions use our original lift. For odd n, there is one fixed point and at most one pair of adjacent circular vertices: each such pair produces a coordinate $+h$ in T. Cutting at the fixed point gives a fully nested matching. Otherwise two disjoint nonempty parts would each give an adjacent pair; every noncrossing perfect linear matching has one, by taking a minimal inner interval. For even n, two points are fixed and there is no adjacent pair: it would require an arrow of length $n - 1 > 2h$ in g. Every pair therefore joins the two arcs between the fixed points; a pair inside one arc would enclose an adjacent leaf. The arcs have equal size and the matching reverses their order.

In both cases σ, and hence T mod n, is a circular reflection. Writing it as $i \mapsto c - i \pmod n$ shows that its shortest arrows reverse two consecutive arcs of sizes $m, n-m$. Every load component has size at most $\lceil n/2 \rceil$, contrary to the retained original packet of the large component. This proves strictness. □

5.7.2 Small invariant blocks

Lemma 5.12. *If the original map preserves consecutive blocks of size at most $\lceil n/2 \rceil$, it has a word of length at most N for any two cursors.*

Proof. First take two invariant blocks of sizes $a, b \geq 2$, $a + b = n$. Put $H = \binom{a}{2} + \binom{b}{2} = N - ab$ and let I be the target inversion count. On vertices $u, \ldots, v$, define a comparison sequence recursively by

$$W[u, v] = (u, u+1, \ldots, v-1)(v-2, v-3, \ldots, u)W[u+1, v-1],$$

with $W[u, v]$ empty for $u \geq v$ and empty sweeps omitted. Each entry is an edge at which to compare. The first sweep places the maximum at v, the return sweep the minimum at u, and recursion sorts the interior. Thus W sorts every list, uses $\binom{k}{2}$ comparisons on k vertices, and moves one step between successive comparisons. For example, $W[0, 3] = (0, 1, 2, 1, 0, 1)$. On a reversed list every comparison exchanges, so W is a reduced word for reversal. Its reverse also sorts every list by Lemma 5.9, since reversal is the maximum in the finite Bruhat order [24, Proposition 2.3.1]. Sort the first block from its middle to the boundary and the second from that boundary to its middle. The extreme internal edges of the blocks are two steps apart, so internal movements total $[\binom{a}{2} - 1] + 2 + [\binom{b}{2} - 1] = H$.

The two boundaries and two orders give endpoint pairs $(A_1, B_0), (B_0, A_1), (A_0, B_1), (B_1, A_0)$, where

$$A_0 = \lfloor (a-2)/2 \rfloor,\ A_1 = \lfloor (a-1)/2 \rfloor, \quad B_0 = a + \lfloor (b-2)/2 \rfloor,\ B_1 = a + \lfloor (b-1)/2 \rfloor.$$

For even n, $B_i = A_i + m$, so the mean approach cost over the four pairs is m. For odd n, one block has a single centre A and the other has centres $A + m, A + m + 1$. Then

$$\min_{v \in \{A+m, A+m+1\}} \operatorname{dist}(x, v) = m - \operatorname{dist}(x, A).$$

Choosing the order gives cost $m - |\operatorname{dist}(s, A) - \operatorname{dist}(t, A)| \leq m$. Exactly I exchanges are made, so the word length is at most $I + H + m$.

Now let L be the largest original block size. For $L = 1$ the map is the identity. For $2 \leq L \leq m$ take $a = L$, $b = n - L$. Then

$$I \leq \sum_j \binom{l_j}{2} \leq n(L-1)/2 \leq L(n-L) - m,$$

since the difference between the last upper bound and its predecessor is $L(n/2 - L) + (n/2 - m) \geq 0$. For $n = 2m + 1, L = m + 1$, we have $I \leq m^2$ and $H + m = m^2 + m$. In either case $I + H + m \leq N$. □

5.7.3 Shallow cycles and adjacent ranks

A finite permutation is *shallow* if $Z = \Delta - I - \mathrm{rk} = 0$ [28, Definition 2]. We use the following literature result: write a shallow cycle starting with its maximum; the remaining word is a separable permutation [28, Corollary 31]. Separable permutations are built by direct and skew sums of singletons: for sizes u, v, these are $(\alpha, \beta + u)$ and $(\alpha + v, \beta)$. Reflecting the order gives the corresponding characterization for a cycle starting with its minimum.

The sum tree has a node with two singleton children, giving adjacent letters of consecutive ranks. Thus a shallow cycle has an arrow between consecutive vertices of its ordered support. Inserting a fixed position at a cut of half-load w increases both Δ and I by $2w$ and preserves reflection length; rank compression of the support preserves Z.

Lemma 5.13. *A nonempty finite permutation without fixed points or unit arrows has $Z \geq 2$.*

Proof. Suppose $Z = 0$ and choose a nonempty load component. Additivity over components and invariant packet gaps gives zero defect to its packet and every filling. Lemma 5.4 then makes the packet a single shallow cycle. If a gap contains active vertices, descend to a nonempty load component of that filling. Its interval is strictly smaller, so this descent terminates at a shallow cycle with no active filling. There are no fixed points, hence its support fills its interval. The preceding shallow-cycle characterization supplies adjacent ranks, which are now adjacent physical positions: a unit arrow, a contradiction. Nonnegativity and evenness of Z give $Z \geq 2$. □

For a finite component with a packet of k cycles define

$$c = Z_{\text{component}} - k + \eta_{\text{in}}, \qquad \eta_{\text{in}} = \sum_{C \text{ in fillings}} (U_C - 1). \tag{5.25}$$

Always $c \geq -1$; $c < 0$ is possible only if $k = 1$ and $Z = \eta_{\text{in}} = 0$.

Lemma 5.14. *If a component has no fixed points or unit arrows, then $c \geq 0$. Equality requires a packet with $k = Z_{\mathcal{P}} = 2$ and no nonempty internal gaps. If arrows of length two are also absent and some internal gap is nonempty, then $c \geq 2$.*

Proof. Lemma 5.4 gives $Z_{\mathcal{P}} \geq 2(k-1)$. If $Z_{\mathcal{P}} = 0$, then $k = 1$: with no gaps Lemma 5.13 would give a unit arrow, while a nonempty gap has $Z \geq 2$, hence $c \geq 1$. If $Z_{\mathcal{P}} > 0, k = 1$, then $Z_{\mathcal{P}} - k \geq 1$. If $k \geq 3$, then $Z_{\mathcal{P}} - k \geq k - 2 \geq 1$. Only $k = 2, Z_{\mathcal{P}} = 2$ remains, and every nonempty gap adds at least two to c. This proves the first part.

We need one further observation. If every arrow of a finite permutation has length at least three and each cycle has half-peak one, then $Z \geq 4$. Take a last packet in the nested decomposition, with empty internal gaps. Its support fills an interval since there are no fixed positions. The arrows of the first three positions are positive and cross the cut after the third position. Its half-load is at least three, so the packet contains at least three cycles. Its defect is at least $2(3-1) = 4$.

If $Z_{\mathcal{P}} > 0$ and a gap is nonempty, $c \geq 2$ follows from $Z_{\mathcal{P}} - k \geq 0$ and $Z_{\text{gap}} \geq 2$. If $Z_{\mathcal{P}} = 0$ and $c \leq 1$, then $k = 1$, $\sum Z_{\text{gap}} = 2$, and $\eta_{\text{in}} = 0$. All filling cycles would have half-peak one, contradicting the preceding observation on a nonempty gap. □

A cycle of half-peak one is bitonic: its positive arrows form a chain from minimum to maximum, and its negative arrows a chain back. Otherwise some cut would have half-load at least two. Such a cycle is shallow, also seen by successively contracting unit rank arrows in the two chains. A packet with at least two cycles has half-peak at least two: at combined peak one, cycle intervals have disjoint interiors and cannot form a strongly connected component.

5.7.4 Combined filling capacity

For a filling of size b put $\kappa = \binom{b}{2} - I + S$, where $S = \sum_C U_C$, and $z = b - 2S \geq 0$. Its half-load profile is bounded by $\min(j, b-j, S)$, hence $\Delta \leq 2S(b-S)$. By (5.5),

$$\kappa \geq \mathrm{rk} + Z + \frac{z(z-1)}{2}, \qquad b \leq 2\kappa + f - 2Z. \tag{5.26}$$

Since $\mathrm{rk} \ge S$, we get $\kappa \ge S + \lfloor z/2 \rfloor = \lfloor b/2 \rfloor$. Thus $b \le 2\kappa + 1$ and $\Delta \le 2\kappa(\kappa+1)$. For a family of gaps with $\sum \kappa_i \le T$ and total size x,

$$\begin{aligned} x &\le 2T + \sum f_i - 2\sum Z_i, \\ \sum \Delta_i &\le 2T(T+1), \\ \sum \Delta_i &\le 2(T - \sum Z_i)^2 \le 2T^2 \quad \text{if } \sum f_i = 0. \end{aligned} \tag{5.27}$$

For the first displacement bound sum the individual bounds and use $\sum \kappa_i^2 \le T^2$. For the second, use $b_i \le 2(\kappa_i - Z_i)$ and $\Delta_i \le b_i^2/2$.

5.8 A large shallow shell

Assume $n \ge 11$, the load is finite, and a large component has an outer packet of defect zero. This packet is a single shallow cycle A. Let its extreme positions be $0, W$, with $W > h$, and its support size $r = 2l + \eta \ge 4$. Write b_i for internal gap sizes and $e = n - 1 - W \ge 2$ for the external interval containing another component. Put $s = n - r = \sum b_i + e$. These intervals are invariant and their fillings are arbitrary. Define

$$\begin{aligned} H &= l(n-l) - \Delta_A \ge 0, \qquad \max \lambda_A = 2U, \\ g &= (b_1, \dots, b_{r-1}, e), \qquad o = \#\{i : g_i \text{ odd}\}, \\ T &= 2\lfloor (s+\eta)^2/4 \rfloor - \sum_i g_i^2. \end{aligned}$$

The same averaging of zero-winding cycles as in Section 5.5 proves $H \ge 0$.

5.8.1 A route through reversed gaps

Reverse each interval g_i and retain A, obtaining $\overline{F} \succeq F$. Exchange adjacent target values $W, W+1$. The outer reversal pair $(W+1, n-1)$ merges with A, giving a root spanning the entire finite window. Its peak remains $2U$; the other cycles are nested pairs. The resulting map G satisfies

$$\Delta_G = \Delta_{\overline{F}} + 2, \quad I_G = I_{\overline{F}} + 1, \quad I_{\overline{F}} + \Delta_{\overline{F}} = 2\Delta_A - r + 1 + \sum_i (g_i^2 - \lceil g_i/2 \rceil).$$

Here $I_A = \Delta_A - r + 1$, and a reversed interval of b positions has $I + \Delta = b^2 - \lceil b/2 \rceil$. There remain $j = (s-o)/2 - 1$ pairs. The two-arc excess for G is $m - U - j - 1 = \lfloor (r+o)/2 \rfloor - U \ge 0$. Internal excesses are no greater: for a reversal, (5.12) gives $K^2 \le b \bmod 2$, and the root load is at least two. The new bridge is empty; the external gap of a root spanning all n positions has excess zero. Service and Lemma 5.9 now give an original word with

$$|w| - N \le E := 3 - U - 2H - p - T. \tag{5.28}$$

5.8.2 Two rank adjacencies and controlled contractions

A shallow cycle without an arrow joining its minimum and maximum has arrows across two distinct adjacent-rank gaps. To see this, write the cycle starting at its minimum; its suffix is separable. Reverse the cycle direction if needed to make its top decomposition a direct sum $\alpha \oplus \beta$. The last, upper part β has length at least two: otherwise the maximum would be last, giving the prohibited extreme arrow. There is a rank adjacency inside β. If $|\alpha| \ge 2$ there is another inside α; if $|\alpha| = 1$, the arrow from rank 0 to rank 1 supplies the second.

Here the extreme arrow is prohibited by $W > h$. Also $r > 2U$: if $r = 2U$, every arrow would cross a maximum-load cut, whereas at most one adjacent-rank gap straddles that cut.

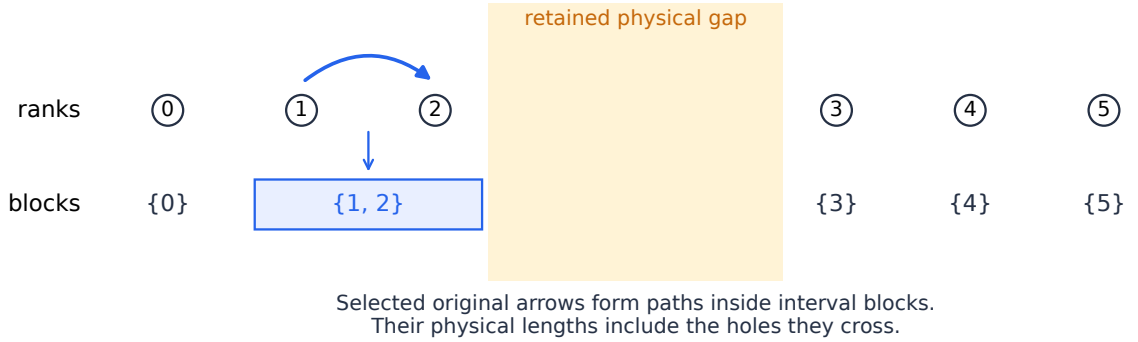


Figure 6: A local rank contraction merges two consecutive support vertices into an interval block. A gap avoided by all selected arrows remains between blocks.

In the following contractions, first replace the ordered support by ranks $0, \dots, r-1$. Record the original physical arrow before each contraction; its physical length is retained when estimating S_{sel}. Contracting a unit rank arrow of a shallow cycle preserves its defect. For an arrow $i \to i+1$, remove the corresponding permutation point and standardize rows and values. If the cut has half-load w, then Δ drops by $2w$, I by $2w-1$, and rank by one. Reflection handles negative arrows. After successive contractions, each vertex represents an interval of original ranks; the selected arrows form a path within it. Arrows between blocks are original arrows with their original signs.

Choose positive thresholds a, z for the two signs and stop when either threshold is reached. At most $k \le a+z-1$ arrows have been selected. If the sign with threshold a is reached, let S_{sel} count only the total original physical length of selected arrows of that sign. The sum of their original rank lengths is at most az. Indeed a block of A_i ranks, with a_i selected arrows of that sign and $k_i = A_i - 1$ selected arrows in all, contributes at most $a_i(A_i - a_i)$, since sources and targets are distinct. Thus

$$\sum_i a_i(A_i - a_i) = a + \sum_i a_i(k_i - a_i) \le a + a(k-a) \le az.$$

At most $\min(a, z)$ of the selected arrows cross a given hole: in a block, arrows of that sign form $A_i - a_i$ monotone paths, and $A_i - a_i \le k - a + 1 \le z$. If the available holes have total length c, we obtain the physical length bound

$$S_{\mathrm{sel}} \le az + \min(a, z)c. \tag{5.29}$$

5.8.3 A dominant internal gap

Figure 6 distinguishes the ranks being contracted from the physical gaps whose lengths remain in (5.29). Suppose $b > s/2$, put $c = s - b - e$, and $v = 2b - s > 0$. Contract only adjacencies avoiding the distinguished gap. After $k \le v-2$ contractions an extreme arrow in the rank cycle would have original physical length at least $r - 1 - k + b \ge n - b + 1 > h$, which is impossible. There are therefore two distinct rank adjacencies, at least one avoiding the gap. Since $b+1 \le h$, we have $v \le r-3$; before each required selection ($k \le v-2$), the current cycle has at least five vertices.

Whenever the stopping thresholds leave $l - j$ original arrows of the attained sign, their original displacement sum and (5.3) give

$$\Delta_A - 2S_{\mathrm{sel}} \le (l-j)(n-l+j), \qquad H \ge j(s+\eta+j) - 2S_{\mathrm{sel}}.$$

This uses the remaining original arrows, rather than a new majorization claim for the contracted cycle. After reflection if needed, the original sign counts are $l+\eta+d, l-d$, with $d \geq 0$. Put $j = \lfloor (v-\eta)/2 \rfloor$. If $j \leq d$, the original sign capacity gives $H \geq d(s+\eta)+d^2 \geq j(v+2e-\eta-j)$. If $j > d$, use thresholds $j+\eta+d$ and $j-d$. Their sum is $2j+\eta \leq v$, so the required contractions are allowed. The sign reaching its threshold has $l-j$ original arrows left. Apply (5.3) to these arrows and then (5.29):

$$\begin{aligned} H &\geq j(s+\eta+j) - 2S_{\text{sel}} \\ &\geq j(v+2e-\eta-j) + 2d(d+\eta) + 2dc. \end{aligned}$$

In either case,

$$H \geq j(v+2e-\eta-j), \qquad 4H \geq (v-\eta)^2 - ((v-\eta) \bmod 2) + 8je. \tag{5.30}$$

If every $g_i \leq s/2$, then $T \geq 0$. For $U = 1$ the cycle is bitonic, $\Delta_A = 2W$, and

$$H = (l-2)^2 - 2 + (l-2)(\eta + \sum b_i) + le \geq 2(e-1) \geq 2.$$

For $U \geq 3$, (5.28) is also paid. Only $U = 2, H = p = T = 0$ remains. If $\eta = 1$, then $T \geq s \geq 2$, a contradiction. If $\eta = 0$, equality $T = 0$ forces exactly two nonempty intervals of equal size: the external e and one internal gap. Equality $H = 0$ forces equal sign counts; removing any one arrow from (5.3) gives $|d_i| \geq e+1$ for every root arrow. But one of the two rank adjacencies crosses an empty gap and is a unit arrow, a contradiction.

For an internal $b > s/2$, (5.28) gives

$$2E \leq 6 - 2U - p - 4H + v^2 - \eta(2s+1). \tag{5.31}$$

Substituting (5.30), for $\eta = 1$ we get $2E \leq 4 - 2U - 4(c+e) - 8je < 0$; for $\eta = 0, j \geq 1$ we get $2E \leq 6 - 2U - 8je < 0$. If $\eta = 0, j = 0$, then $v = p = 1$ and $2E \leq 6 - 2U - 4H$. The cases $U \geq 3$ and $U = 1$ are already paid. For $U = 2, H = 0$, the original prefix bound requires $|d_i| \geq \lceil (s+1)/2 \rceil > c+1$. Yet a rank adjacency avoiding the dominant gap has length at most $c+1$, again a contradiction.

5.8.4 A dominant external interval

Suppose $e > c := \sum b_i$, and put $t = e - c \geq 1$. Since $W > h$, $t \leq r - 2 - p$. Now contract any unit rank arrow. If the threshold sum is at most t, stopping takes at most $t - 1 \leq r - 3$ contractions, so unit rank arrows do not run out. Use sign counts $l+\eta+d, l-d$, $d \geq 0$, as above. With $j = \lfloor (t-\eta)/2 \rfloor$, the preceding sign argument, applied to all holes of total length c, gives the following. When $j > d$, $S_{\text{sel}} \leq (j+\eta+d)(j-d) + (j-d)c$, so

$$H \geq j(t-\eta-j) + 2d(d+\eta) + 2dc \geq \lfloor (t-\eta)^2/4 \rfloor.$$

When $j \leq d$, the sign-capacity bound $H \geq d(s+\eta) + d^2$ is at least the same quantity. Hence

$$H \geq \lfloor (t-\eta)^2/4 \rfloor. \tag{5.32}$$

We have $2E \leq 6 - 2U - p - 4H + t^2 - \eta(2s+1)$. For $\eta = 1$ this gives $2E \leq 4 - 2U - 4c$; for $\eta = 0$ it gives $2E \leq 6 - 2U$. Only $U = 1$, or $\eta = 0, U = 2$, remains.

For $U = 1$, $\Delta_A = 2(r+c-1)$. If $r = 2l$, $l \geq 2$, then

$$E \leq -2l^2 + 8l - 2 - 4(l-1)c - 2lt + t^2/2 - p/2.$$

The function $t^2/2 - 2lt$ decreases on $1 \leq t \leq 2l-2$. For $l \geq 3$ even its value at $t = 1, c = 0$ is negative. For $l = 2$ only $t = 1, 2$ occur. At $t = 1$, $e \geq 2$ implies $c \geq 1$; at $t = 2$, substitution gives $E \leq 0$. If $r = 2l+1$, $l \geq 2$, then

$$E \leq 3/2 - 2l^2 + 6l - (4l-2)c - (2l+1)t + t^2/2 - p/2.$$

Here $t \le 2l-1$. For $l \ge 3$ it suffices to take $t = 1, c = 0$. For $l = 2, t = 1$ we have $c \ge 1$; for $t \ge 2$, use $t = 2, c = 0$. All these values are nonpositive.

Finally suppose $r = 2l, U = 2$. Put $K_* = H - \lfloor t^2/4 \rfloor \ge 0$. Exactly,

$$E = 1 - 2K_* - (c^2 - \sum b_i^2).$$

If $E > 0$, then $K_* = 0$ and at most one internal gap is nonempty. The root displacement is at most $4(r + c - 2)$, hence

$$K_* \ge l^2 - 8l + 8 + lt - \lfloor t^2/4 \rfloor + (2l - 4)c.$$

The expression $lt - \lfloor t^2/4 \rfloor$ strictly increases on $1 \le t \le 2l - 2$. For $l \ge 6$, the lower bound is at least $l^2 - 7l + 8 \ge 2$. For $l = 5, t = 1$, we need $c \ge 1$, giving $K_* \ge 4$; for $t \ge 2$, $K_* \ge 2$. The value $l = 2$ is excluded by $r > 2U$. For $l = 3$, only $(t, c) = (1, 1), (1, 2), (2, 0), (2, 1), (3, 0)$ remain; for $l = 4$, only $(1, 1), (2, 0)$ remain. At $t = 1$ we have $H = 0$: the original prefix bound requires $|d_i| \ge c + 1 > 1$, but the single nonempty gap and the two rank adjacencies force a unit arrow. All other cases have $n = 8, 9, 10$, contrary to $n \ge 11$.

Proposition 5.15. *For $n \ge 11$, a large finite component with a shallow outer packet and arbitrary fillings is paid within length N for every pair of cursors.*

5.9 Exactly two finite components

Let $n \ge 11$. If both components are small, apply Lemma 5.12; otherwise there is a unique large component. Proposition 5.15 handles a large packet of defect zero. It remains to consider positive defect of the large packet. Let $g \le g_*$ be the separator widths, $a = g + g_* - 2 \le f$, and $v \ge 2$ the size of the smaller component. For every internal gap of either original packet put $\gamma_i = \binom{b_i}{2} - I_i$. Retain the large packet, reverse its gaps, and reverse the entire external interval of length $v + a$. Lemma 5.11 and additivity give

$$\delta \ge \sum_i \gamma_i + av + a(a-1)/2 + 1. \tag{5.33}$$

To include gaps of the smaller component, use $I_{\text{other}} + \sum_{i\ \text{other}} \gamma_i \le \binom{v}{2}$. The positive-defect large packet has at least four vertices; Lemma 5.4, including the other component's contribution, gives $\delta \ge 2$.

For the component contributions (5.25),

$$U + c_{\text{big}} \ge 2, \qquad c_{\text{other}} \ge -1. \tag{5.34}$$

For $k \ge 2$ use $Z_{\mathcal{P}} - k \ge k - 2 \ge 0$ and $U \ge 2$. For $k = 1$, positive defect is at least two, and the peak is not one, since a cycle of half-peak one is shallow.

5.9.1 All gaps

Lemma 5.2 gives $\varepsilon \ge f - 1$. If $\delta \ge 3$, then $\gamma_i \le \delta - 1$ and Lemma 5.10 imply $K_i \le \delta - 1$: $K_i \ge \delta$ would give $\delta(\delta - 1) \le 2(\delta - 1)$. Therefore

$$2g + K_i - w_i \le \delta + 2g - 3 \le \delta + \varepsilon,$$

since $2g - 2 \le a \le f$. If $\delta = 2$, (5.33) forces $a = 0$, so $g = g_* = 1$, $\gamma_i \le 1$, and $K_i \le 2$; the old-gap excess is at most $2 \le D$.

The unpaid external excess satisfies $E_{\text{ext}} \le a + g + 1 - \delta - \varepsilon$. Using $2g \le a + 2$, $v \ge 2$, and (5.33), we obtain

$$2E_{\text{ext}} \le 2 - a^2 - 2\varepsilon.$$

If $(a, \varepsilon) \ne (0, 0)$, integrality gives $E_{\text{ext}} \le 0$. If $a = \varepsilon = 0$, then $g = 1$ and $E_{\text{ext}} \le 2 - \delta \le 0$. A new bridge costs no more than the external arc.

5.9.2 The two-arc peak and the single activity exception

Rewrite (5.22) using component contributions:

$$P = 2g - U - c_{\text{big}} - c_{\text{other}} - A - 1 \le 2g - A - 2. \tag{5.35}$$

The activity bounds give $A \ge 2g - 2$, except for

$$f = a = 2, \quad p = 0, \quad \varepsilon = 1, \quad g = g_* = 2, \quad A = 0. \tag{5.36}$$

Here is the exhaustive check. For $f \ge 4$, $\varepsilon \ge f$ and $A \ge f$; for $f \le 1$, $g = 1$; for $f = 3$, $A \ge 2$ and $g \le 2$. For $f = 2, p = 1$, the parity of ε and the activity bounds give $\varepsilon \ge 2$. For $f = 2, p = 0$, only (5.36) remains.

In this exceptional tuple there are no fixed positions inside components, and both sign counts are $m - 1$: a smaller count at most $m - 2$ would give $\Delta \le m^2 - 4$, whereas $\Delta = m^2 - 1$. A unit arrow, together with the prefix of the other arrows of that sign, would require $\varepsilon \ge 2^2 - 2 = 2$. Thus unit arrows are absent. By Lemma 5.14, both component contributions are nonnegative. If $P > 0$, (5.35) forces $U = 2, c_{\text{big}} = c_{\text{other}} = 0$. The same lemma excludes internal gaps in both components. Their sizes L_1, L_2 satisfy $L_1 + L_2 = 2m - 2$, and

$$\Delta \le 4(L_1 - 2) + 4(L_2 - 2) = 8m - 24 < m^2 - 1,$$

since the difference is $(m - 4)^2 + 7$. This is a contradiction. Every two-component target is therefore paid for $n \ge 11$.

5.10 At least three components: a finite cutoff

After Lemma 5.12 and Proposition 5.15, there remains a unique large component of size $L > \lceil n/2 \rceil$ whose packet has positive defect. Let the other $t = \rho - 1 \ge 2$ components have sizes $l_j \ge 2$, put $v = \sum_j l_j$, and let a count fixed positions outside all components. Then

$$L = 2m + p - a - v, \quad v + a \le m - 1, \quad B \le a + t - \mathbf{1}_{\{a>0\}}. \tag{5.37}$$

The last bound follows from $\sum g_j = a + t + 1$ and $g_* \ge 1 + \mathbf{1}_{\{a>0\}}$.

5.10.1 Inversion reserve between outside components

Retain the large packet, reverse its internal gaps, and reverse the entire external interval of size $v + a$. The strict part of Lemma 5.11 gives

$$\delta \ge \sum_{\text{all}} \gamma_i + M_{\text{actual}} + 1, \qquad M_{\text{actual}} = \binom{v+a}{2} - \sum_j \binom{l_j}{2} \ge M,$$
$$M = 2(t-1)(v-t) + av + a(a-1)/2. \tag{5.38}$$

The sum includes internal gaps of all packets. Reversing gaps inside an outside interval cannot exceed its full capacity $\binom{l_j}{2}$. For the last inequality write $l_j = 2 + x_j$, $x_j \ge 0$; the discarded remainder is $\sum_{i<j} x_i x_j$. Also

$$M - 2(a + t) = 2(t-1)(v-2t) + 2t(t-2) + a(v-2) + a(a-1)/2 \ge 0.$$

By Lemma 5.10, old gaps have excess $2B + K_i - w_i \le 2B + \gamma_i - 1 \le \delta - 1 \le D$. The external excess is $2B + g_* - 1 = B + a + t \le 2(a + t) \le \delta$. New bridges cost no more. Thus only (5.22) remains, with necessary bounds

$$P \le 2B + t - U - A, \qquad P \le 2B + m - U - 2 - M - \varepsilon. \tag{5.39}$$

5.10.2 Density, area, and short outside arrows

Let R_0 be the number of active vertices of the large packet. We have $R_0 \geq 4$: a two- or three-cycle would have an extreme arrow of length $L-1 > h$, while two nontrivial cycles require at least four vertices. Take all nonempty internal gaps of the large packet and all t outside components. Denote their lengths by s_i, including singleton gaps, and put

$$r = R_0 + a, \quad V = \sum_i s_i = n - r, \quad S_2 = \sum_i s_i^2, \quad H_0 = V^2 - 2S_2.$$

Each length is at most $m-1$, $V \leq 2(m-1)$, and at least t lengths are at least two. The bound on V follows from $R_0 \geq 4$.

For a finite interval of size s_i with f_i fixed points, $2\Delta_i \leq s_i^2 - f_i^2$: the smaller sign count is at most $(s_i - f_i)/2$. Remove the arrows of all these intervals from the original sign prefixes. If y_+, y_- are m minus the remaining sign counts, then

$$\varepsilon + \Delta_{\text{intervals}} \geq y_\pm(y_\pm + p), \qquad y_+ + y_- = V + a - p.$$

Adding and completing the square gives $4\varepsilon + p \geq (V+a)^2 - 4\Delta_{\text{intervals}}$, whence

$$2A \geq H_0 + 2(V-1)a + a^2 + 2\sum_i f_i(f_i - 1) + p \geq H_0 + 2(V-1)a + a^2 + p. \tag{5.40}$$

The discarded sum is nonnegative since every f_i is an integer.

The large packet's half-profile is bounded by $\min(j, L-j, U)$. Even if the maximum U occurs in another component, $2U \leq L$, since that component is smaller. Thus

$$\Delta_{\text{core}} \leq 2U(L-U), \qquad 2(Q-\varepsilon) \leq 4U(L-U) + S_2. \tag{5.41}$$

Each outside component has an arrow of each sign of length at most $l_j - 1$. Removing one of each, with $x = m - \min(k_+, k_-) \geq 0$, gives

$$\varepsilon \geq (x+t)(x+t+p) - 2(v-t), \qquad f \leq 2x + p.$$

Since $t \geq 2$, the expression $x^2 + px + (2t-4)x + p(t-2)$ is nonnegative. Therefore

$$\varepsilon \geq 2f + t^2 - 2(v-t). \tag{5.42}$$

5.10.3 Absolute bounds for an unpaid case

Suppose $P \geq 1$. From (5.39), the activity bounds, and (5.42), we obtain

$$\begin{aligned} &2\varepsilon + p \leq 3f + 3t - 2, && (f-3)^2 \leq 6t + 5 - p, && \\ &f \leq t + 5, && \varepsilon \leq 3t + 6, && U \leq 3t + 1, \\ &4v \geq 2t^2 + t + 2. && && \end{aligned} \tag{5.43}$$

The first row follows from $A \leq 2a + 3t - 2 \leq 2f + 3t - 2$ and $f^2 \leq 4\varepsilon + p$. If $f \geq t+6$, the square bound contradicts $(t+3)^2 > 6t + 5$. The bound on ε follows. If $a = 0$, $U \leq 3t - 1$. If $a > 0$, then $U \leq 3f + 3t - 3 - 2\varepsilon - p$, so $2U \leq 6t + 3 - (f-3)^2 - p \leq 6t + 3$. The last row compares the upper and lower bounds on 2ε.

For $t \geq 3$, integrality gives $2v \geq t^2 + 3$. The second bound in (5.39) and (5.37) give $m \geq 2(t-1)v - 2t^2 + 4$. Hence

$$m \geq t^3 - 3t^2 + 3t + 1 \quad (t \geq 3). \tag{5.44}$$

Since $V \geq 2t \geq 4$, (5.40) and $A \leq 2a + 3t - 2$ give $H_0 \leq 6t - 3$.

Either $m \le 42t+23$, or $m \ge 42t+24 \ge 12U+2\varepsilon$. In the latter case, $S_2 \le (m-1)V$ and (5.41) imply

$$r \le 8U-2-p \le 24t+6-p. \tag{5.45}$$

Here is the algebra. Weaken $L \le n$ in the area bound to get

$$(m-1)(r-(8U-2-p))+4U^2-8U-4Up+2+2p-2\varepsilon \le 0.$$

If $r \ge 8U-1-p$, the left side is at least $m+4U^2-8U-4Up+1+2p-2\varepsilon > 0$, since $m \ge 12U+2\varepsilon$. This proves (5.45).

Lemma 5.16 (Square envelope)**.** *Suppose the lengths sum to $V \le 2M_0$, their maximum is $x \le M_0$, and $S_2 \le x^2+(V-x-w)^2+w^2$ for $w \ge 0$. Then*

$$S_2 \le M_0^2+(V-M_0-w)^2+w^2, \qquad H_0 \ge 4M_0w-(2M_0-V+2w)^2. \tag{5.46}$$

Proof. If $x \ge V-M_0-w$, use convexity of $z^2+(V-z-w)^2+w^2$ on $[V-M_0-w, M_0]$. Its endpoint values coincide, so its value at x is at most that value. If $x < V-M_0-w$, use $S_2 \le xV \le (V-M_0-w)V$. The proposed upper bound minus the latter expression is $(M_0+w)(2M_0-V)+2w^2 \ge 0$. Expanding squares gives the second bound. □

Take $M_0 = m-1$. If at least three intervals are nonempty, use $w=1$: after removing the largest, at least two positive integer parts remain. If at least $t-1$ lengths of size at least two remain, use $w = 2(t-2)$. Concentrate the remaining mass in one part, leaving $t-2$ parts of size two; this gives the stronger bound $(V-x-w)^2+4(t-2) \le (V-x-w)^2+w^2$.

For $t=2$, the large branch begins at $m \ge 108$. Equation (5.45) gives $R_0 \le 54-p < L$, so there is a nonempty internal gap as well as two outside intervals. With $w=1$ we get $H_0 \ge 4(m-1)-(r-p)^2$. Since $H_0 \le 9$, this implies $m \le 732$.

For $t \ge 3$, removing the largest interval leaves at least $t-1$ outside lengths. With $w = 2(t-2)$ and (5.45),

$$8(t-2)(m-1) \le (28t-4)^2+6t-3. \tag{5.47}$$

Together with (5.44), this would require $8t^4-40t^3-712t^2+170t-13 \le 0$. At $t = 13+z$, $z \ge 0$, its left side is $8z^4+376z^3+5840z^2+31682z+22477 > 0$. In the small branch $m \le 42t+23$, the contradiction is $t^3-3t^2-39t-22 > 0$ for $t \ge 13$; after $t = 13+z$, all coefficients are positive. For $t = 3, \dots, 12$, the maxima of the small bound and (5.47) are, respectively, $802, 731, 772, 842, 923, 1010, 1100, 1192, 1285, 1379$. Thus

$$P \ge 1 \quad \Longrightarrow \quad 2 \le t \le 12, \quad m \le 1379, \quad n \le 2759. \tag{5.48}$$

This analytic cutoff is proved before any computation is used.

6 The complete finite certificate

Appendix E contains the full program. We now justify its ranges and each kind of rejection. The code first enumerates necessary integer parameters, then the small actual outside components.

6.1 Scalar parameters and necessary filters

The code accepts a threshold $k=1$ or $k=2$; the proof uses $k=1$. If $P \ge k$, replace the actual B by its upper bound: every estimate used below is thereby weakened. Put

$$B = a+t-\mathbf{1}_{\{a>0\}}, \qquad M = 2(t-1)(v-t)+av+a(a-1)/2,$$
$$L = 2m+p-a-v, \qquad x = L-R_0, \qquad h_0 = m+p-2,$$
$$T = 2B+m-U-2-M-\varepsilon-k, \quad E_2 = (v-2t+2)^2+4(t-1).$$

Equation (5.38) and the definition of P give $\sum_{\text{all}} \kappa_i \le T$, in particular $T \ge 0$. Here $\kappa_i = \gamma_i + S_i$ ranges over internal gaps of all packets. Besides (5.43)–(5.48), we need even $Q-\varepsilon$, $x \le 2T+f-a$, and

$$\begin{aligned} \sum_{\text{big}} b_i^2 &\le \min\{J(x), 4T^2+2T+x\}, \\ 2\Delta_{\text{external}} &\le E_2 - (v \bmod 2), \\ 2\Delta_{\text{big fill}} &\le 4T\big(T + \mathbf{1}_{\{f>a\}}\big), \\ (x+v+a)^2 &\le 4\varepsilon + p + 4(\Delta_{\text{external}} + \Delta_{\text{big fill}}), \end{aligned} \tag{6.1}$$

where $J(x) = x^2$ for $0 \le x \le h_0$, and $J(x) = h_0^2 + (x-h_0)^2$ for $h_0 \le x < 2h_0$. Indeed $x \le L-4 < 2h_0$. Concentrating lengths subject to $b_i \le h_0$ gives J. Since $\lfloor b_i/2 \rfloor \le \kappa_i$, we have $b_i^2 \le 4\kappa_i^2 + 2\kappa_i + b_i$; summation gives the other square bound. Outside lengths are at least two, hence $\sum l_j^2 \le E_2$ and $2\Delta_{\text{external}} \le E_2 - (v \bmod 2)$. The remaining estimates are (5.27) and the original sign prefixes used to derive (5.40).

The program loops over the following ranges:

$$\begin{gathered} 2 \le t \le 12, \quad p \in \{0,1\}, \quad 0 \le f \le t+5, \quad 0 \le \varepsilon \le 3t+6, \\ 0 \le a \le f, \quad 1 \le U \le 3t+1, \quad 2t+a+1 \le m \le 1379, \quad 2t \le v \le m-a-1, \quad 0 \le x \le L-4. \end{gathered}$$

For $t \ge 3$, (5.44) strengthens the lower bound on m. Equation (5.42) strengthens the lower bound on v, and $T \ge 0$ its upper bound.

Exact square-root bounds accelerate the loop over x while preserving all admissible rows. If

$$\begin{aligned} B_0 &= 4T^2+2T, & E_0 &= E_2 - (v \bmod 2), \\ E_1 &= 4T(T + \mathbf{1}_{\{f>a\}}), & W &= 2(Q-\varepsilon) - 4U(L-U), \end{aligned}$$

then necessarily

$$\begin{aligned} J(x) + E_2 &\ge W, & x + B_0 + E_2 &\ge W, \\ (x+v+a-1)^2 + 2v - 1 + p - 2(B_0+E_2) &\le 2A, \\ (x+v+a)^2 &\le 4\varepsilon + p + 2E_1 + 2E_0. \end{aligned}$$

These follow directly from (5.41), (5.40), and (6.1), and give the integer square roots in the code. The square argument is nonnegative on each branch of J; every remaining x is tested against the full density bound (5.40). Duplicate rows from the two branches are removed by a set.

6.2 All outside permutations and the sign tests

For each scalar row the program checks $v - 2(t-1) \le 5$. This is an outcome of the complete scalar enumeration, not an imposed restriction. Thus every outside component has size two through five. The function `make_catalogue` enumerates all permutations of those sizes and retains exactly those with connected load. The function `describe` computes cycles, passages, the root packet, and its load independently from the permutation.

For an outside family compute $I_e, \Delta_e, S_e, c_e, f_e$. Here S_e includes only fillings of outside packets, and c_e sums the component contributions (5.25). Put $f_b = f - a - f_e$, the number of fixed points in the large component. The exact form of (5.22) and strict inversion capacity require

$$\begin{aligned} c_{\text{big}} &\le 2B - U - c_e - A - 1 - k, \\ \sum_{\text{big}} \kappa_i &\le 2B + m - U - 2 - \varepsilon - k - \Big[\binom{v+a}{2} - I_e\Big] - S_e. \end{aligned} \tag{6.2}$$

Filling length, displacement, and density are tested again using these exact outside quantities. The restriction $R_0 \geq 2U$ is used only when no outside packet attains U. Rounding the displacement capacity down to $2\lfloor S_2/4 \rfloor$ is valid: finite displacement is even and at most $S_2/2$.

For a sign s, put $x_s = m - k_s$, where k_s counts its arrows. Then $x_+ + x_- = f - p$. If r arrows of that sign have been selected, with total absolute length b, majorization of the remaining arrows gives

$$\varepsilon \geq (x_s + r)(x_s + r + p) - 2b. \tag{6.3}$$

The function `signs_ok` tests every prefix of the sorted absolute outside arrow lengths and every possible x_s. The inequality $x_s(x_s+p) \leq \varepsilon$ implies $-\varepsilon-2 \leq x_s \leq \varepsilon+2$, so the range is exhaustive. Adding a hypothetical arrow of length one or two tests whether such an arrow can occur in the large component. If both signs fail, that length is excluded.

After these necessary tests, the code uses only the two parts of Lemma 5.14 and the explicit route below:

Branch	Mathematical basis
`creditCuts`	$f_b = 0$, no unit arrows: $c \geq 0$; $c = 0$ requires $x = 0$.
`bitonicCuts`	No lengths $0, 1, 2$, $x > 0$: $c \geq 2$.
`tourCuts`	The direct three-port route below.

Thus every rejection has a necessary mathematical justification. No independent global majorization is assumed for the outside maps.

6.3 Enumeration output and the remaining routes

For $k = 1$ there are 757 distinct scalar rows and 13 009 outside families. The geometric and sign tests retain 691 families. The first part of Lemma 5.14 rejects 682, and its second part rejects seven. The two remaining rows have

$$n \in \{16, 20\}, \quad p = f = \varepsilon = a = 0, \quad U = 4, \quad x = 0, \quad c_{\text{big}} = 0,$$

with two adjacent unit pairs outside. By rigidity the large packet has $k = Z = 2$ and no gaps. Altogether $q = 4$, $Z = 2$, and $A = 0$, so $D = m - 2$. Since $a = 0$, place the two outside pairs at positions $0, \ldots, 3$; their ports are $0, 2$. Serving the large root at a peak port costs $\Delta_{\text{core}} - 8$ rotations, while each pair needs zero internal rotations.

For $n = 16, 20$, `three_port_max` checks every position of the third port and both cursors. For each of the six orders of the three ports, it adds shortest circular distances. The maximum over cursor and port positions of the minimum of these six lengths is n. These are explicit admissible walks; they need not be optimal among all walks. After redundant X letters are deleted,

$$|w| \leq C + \Delta - 12 + n = N - (m - 2) - 12 + 2m = N + m - 10 \leq N.$$

The number of untreated rows, `survivors`, is zero. Together with the already paid gaps, this completes $\rho \geq 3$.

6.4 Reproducibility and exact arithmetic

The program uses the C++17 standard library and signed 64-bit integers. Assertions must remain enabled. In the proved ranges, square-root radicands are less than $3 \cdot 10^7$; even the square of an intermediate binary-search value is less than $9 \cdot 10^{14} < 2^{63}$. All other products have absolute value below 10^{16}. If the numerator of the upper bound on v is negative, there is no admissible positive v, so that pass is empty.

For small sizes, BFS enumerates all $n!$ permutations for $4 \leq n \leq 10$ using bijective Lehmer ranks, checking all three outgoing edges. It asserts that exactly $n!$ states were reached. The diameters are

n	4	5	6	7	8	9	10
D_n	6	10	15	21	28	36	45

This checks ordinary permutations; it makes no bound assertion for an arbitrarily prescribed affine lift at small n.

7 Completion of the proof and reproduction

Proof of Theorem 1.1. BFS covers $4 \le n \le 10$. Let $n \ge 11$ and choose an arbitrary target list. Choose the lift and final phase c as in Section 5.1. They satisfy the original constraints (5.3). If the load is connected, apply Theorem 5.8. Otherwise a zero edge gives a common finite window. Lemma 5.12 covers small components; Proposition 5.15 covers a large shallow component; Section 5.9 covers two components; and Sections 5.10–6 cover all remaining cases. A zero lift needs only a shortest rotation, at most $m \le N$ letters.

Every construction for $n \ge 11$ allows arbitrary initial and final physical cursors. Take them to be 0 and the chosen c. Token i ends at physical position $p_i + c$, so reading from cursor c gives exactly the target list, without an extra rotation. Thus $D_n \le N$. Theorem 4.1 gives the reverse inequality. The sizes $n = 1, 2, 3$ are checked directly. □

To reproduce the finite part, save the complete C++17 program from Appendix E as `upper_bound.cpp`, then run:

```
c++ -O3 -std=c++17 upper_bound.cpp -o /tmp/lrx-certificate
/tmp/lrx-certificate 1
```

Assertions must remain enabled: do not use `-DNDEBUG`. Compare the output with the complete reference output printed after the program in Appendix E. The final refinement check must report `survivors=0`.

8 Formalization

We have also formalized the proof of the main result in Lean 4. It is available at the repository [33] and was created with the help of our agentic system discussed in the Introduction.

9 Acknowledgements

We thank the members of the CayleyPy collaboration for useful related discussions. F.L.-M. also thanks A. Guerrieri and B. Stefanski for discussions. The work of F.L.-M. was supported by the STFC grants APP69281 and ST/C004225/1. F.L.-M. is grateful to Anthropic for access to their models through the Claude Team plan for scientists.

A Details of the independent lower-bound proof

This appendix proves all steps used in Section 4. The triangle-parity observation is shared with the argument in [17, Section 4.2]; the two-colour rule is a standard balance criterion. Both are proved below in our notation, followed by the physical cursor argument.

Plan. Take a shortest word v from e to p^*, with q letters X (exchanges) and ρ letters L, R (rotations), $|v| = q + \rho$. We show: (1) no pair of cards is exchanged twice; (2) the cards split into two groups forming arcs, of sizes a and b, and exactly the pairs inside a group are exchanged, so $q = \binom{a}{2} + \binom{b}{2}$; (3) there are at least ab rotations. Then $|v| \ge \binom{a}{2} + \binom{b}{2} + ab = \binom{n}{2}$.

Below, "the pair exchanged by a letter X" means the two *labels* standing first when this letter is executed.

Detailed proof of Theorem 4.1. Step 1 (no pair is exchanged twice). Let $v = uXzXt$ where both marked letters exchange the same labels a, b. The moves act on places and do not look at labels, so they commute with the relabelling $\tau : a \leftrightarrow b$. Both marked letters act as τ and can be cancelled: uzt has the same endpoint and is shorter by 2, contradicting minimality.

Step 2 (triangle parity and two colours). The triangle-parity observation is used in [17, Section 4.2]; the resulting two-colour rule is Harary's balance theorem [30, Section 1]; see [31, Section 5.2] for a proof. We retain the direct argument in our notation. Let $E(a, b) = 1$ if the pair a, b is exchanged (by Step 1, exactly once) and $E(a, b) = 0$ otherwise. For any three labels

$$E(a, b) + E(a, c) + E(b, c) \equiv 1 \pmod 2.$$

Indeed, look at the cyclic order of the three labels on the circle. A rotation does not change it; an exchange of two labels of the triple reverses it; an exchange with another label does not change it. The passage $e \mapsto p^*$ reverses the cyclic order of every triple, so every triple sees an odd number of exchanges.

This gives a 2-colouring: fix a label z, give it colour 0 and give $a \neq z$ the colour $1 - E(z, a)$. The triangle rule implies that $E(a, b) = 1$ exactly for pairs of the same colour. If the colour classes have a and b labels ($a + b = n$), then

$$q = \binom{a}{2} + \binom{b}{2}.$$

Step 3 (each colour occupies an arc). Suppose two labels of one colour are separated on both sides by labels of the other colour. Every exchange is monochromatic, so the colour at each physical place is invariant. Thus the separating arcs remain in place, and the two labels never become neighbours. Yet they must be exchanged. Hence each colour class is a contiguous arc of the circle.

Step 4 (degenerate cases). If one class is empty, then $q = \binom{n}{2} = N$ and we are done. Suppose a class has one label. A shortest word contains no XX, so a rotation separates any two letters X: $q \leq \rho + 1$, and $|v| \geq 2q - 1 = (n - 1)(n - 2) - 1$, which is at least N for $n \geq 5$. For $n = 4$ use parity: each of L, R, X is an odd permutation of four elements, while $p^* = (1, 0, 3, 2)$ is even; so the length is even, and $|v| \geq 5$ gives $|v| \geq 6$.

Step 5 (exchanges stay inside the arcs). Let arc A occupy places $0, \ldots, a - 1$ and arc B places $a, \ldots, n - 1$. The two edges between the arcs carry labels of different colours, so X is never applied there: every exchange swaps neighbouring places inside one arc. At the end the order of labels inside each arc is reversed, and the initial and final cursor positions c_0, c_1 satisfy $c_0 + c_1 + 2 \equiv a \pmod n$. Indeed, the target p^* sends a visible position i to label $1 - i \pmod n$, so in physical coordinates its reflection is $z \mapsto c_0 + c_1 + 1 - z$. Reversal inside A is $z \mapsto a - 1 - z$, giving the congruence.

Step 6 (counting exchanges). By the classical inversion-length formula [24, Proposition 1.5.2], reversing an arc of a labels by adjacent exchanges takes $\binom{a}{2}$ exchanges, so $q \geq \binom{a}{2} + \binom{b}{2} = M$. Since $M + ab = N$, it remains to show that there are at least ab rotations.

Step 7 (counting rotations). Now $a, b \geq 2$. Write $C \in \{0, \ldots, n - 1\}$ for the physical cursor and $H \in \mathbb{Z}$ for its winding number, increased when L crosses $n - 1 \to 0$ and decreased for the reverse crossing. Project to the two arcs by

$$(c_A, h_A) = (\min(C, a - 1), H), \qquad (c_B, h_B) = \begin{cases} (b - 1, H - 1), & C < a, \\ (C - a, H), & C \geq a. \end{cases}$$

For L, edges $0, \ldots, a - 2$ advance projection A, and the edge $n - 1$ closes its projected circle, taking $(a - 1, H)$ to $(0, H + 1)$. Edges $a - 1, \ldots, n - 2$ similarly advance projection B; at the first

of these edges its state changes from $(b-1, H-1)$ to $(0, H)$. In each case the other projection is unchanged; R reverses these steps. Thus every physical rotation makes exactly one projected cursor step, and $\rho = \rho_A + \rho_B$. In a projected arc of size j let z_x be the local place of label x and set $F_x = z_x - \mathbf{1}_{z_x > c} + h$. An internal exchange preserves all F_x; a projected rotation changes exactly one coordinate by 1 or -1. Indeed, an internal exchange at $c, c+1$ gives both positions the same folded coordinate $c + h$. A forward step changes only the label at the new cursor; on the step $j-1 \to 0$, the increase of h cancels the indicator change for every label except the label at 0. Consequently a projected path from state u to state v costs at least $\sum_x |F_x(v) - F_x(u)|$ rotations.

Here the external cursor is $c = j-1$, where $F_x = z_x + h$. Write $C_0, C_1 \in \{0, \dots, n-1\}$ for the physical endpoint cursors. Step 5 gives $C_0 + C_1 + 2 = a$ or $a+n$. In the first case both endpoints are internal to A, with $c_{A,0} + c_{A,1} + 2 = a$, and both endpoints of B are external. In the second case both endpoints of A are external. If neither physical endpoint is $a-1$, both are at least a and $c_{B,0} + c_{B,1} + 2 = b$. If one is $a-1$, the other is $n-1$, so both endpoints of B are external too. Thus each projected arc either has both endpoints external, or has $c_0 + c_1 + 2 = j$. In the latter case $F_x(0) + F_x(1) = j - 2 + h_0 + h_1 =: K$, independently of x, since $z_x(1) = j - 1 - z_x(0)$ and the two indicators sum to one: $\mathbf{1}_{\{z_x(0) > c_0\}} + \mathbf{1}_{\{z_x(0) \le c_0\}} = 1$. Moreover, its path visits the external cursor: the other arc is nontrivially reversed, so an exchange must occur there, at which moment this projection is external. At such a state m, the two path segments therefore cost at least

$$\sum_x \big(|F_x(m) - F_x(0)| + |F_x(1) - F_x(m)|\big) \ge \sum_x |2z_x(m) - (K - 2h_m)| \ge \lfloor j^2/2 \rfloor.$$

The last inequality is the median minimization property [34, p. 38], applied to $0, 2, \dots, 2j-2$; the $z_x(m)$ run through $0, \dots, j-1$. Its evaluation as $\lfloor j^2/2 \rfloor$ is retained in Appendix B.3. If both endpoint cursors are external, the direct endpoint distance is $\sum_{z=0}^{j-1} |j - 1 - 2z + h_1 - h_0|$, so the same median bound applies.

Thus $\rho \ge \lfloor a^2/2 \rfloor + \lfloor b^2/2 \rfloor$. If both sizes are odd, n is even and Step 5 makes $c_1 - c_0$ odd. Each rotation changes cursor parity, so ρ is odd, whereas the two floors are even; this adds 1. In all cases

$$\rho \ge \Big\lfloor \frac{a^2 + b^2}{2} \Big\rfloor \ge ab,$$

because $a^2 + b^2 - 2ab = (a-b)^2 \ge 0$.

Hence $|v| = q + \rho \ge \binom{a}{2} + \binom{b}{2} + ab = N$. □

B Proofs and identifications for standard ingredients

This appendix recalls standard results and records their application to our conventions. Sources are cited at each use in the main text; the checks involving the physical cursor remain part of the application.

B.1 Finite inversion and transposition lengths

For adjacent transpositions we use [24, Proposition 1.5.2]. One adjacent exchange changes $I(q)$ by exactly 1, so a word from the identity to q needs at least $I(q)$ exchanges. If $I(q) > 0$, some adjacent pair is decreasing; exchanging it lowers I by 1. Iteration sorts q in exactly $I(q)$ steps. For a reversed interval of size j, every pair is inverted, so $I = \binom{j}{2}$. In Appendix A, step 6, this is applied separately to the two arcs.

The unrestricted transposition length $T = n - c(q)$, with fixed points included in $c(q)$, is the standard cycle-count formula used in [26, Section 2]. A transposition changes the number of cycles by one, so at least $n - c(q)$ are necessary. A cycle of length r is a product of $r - 1$

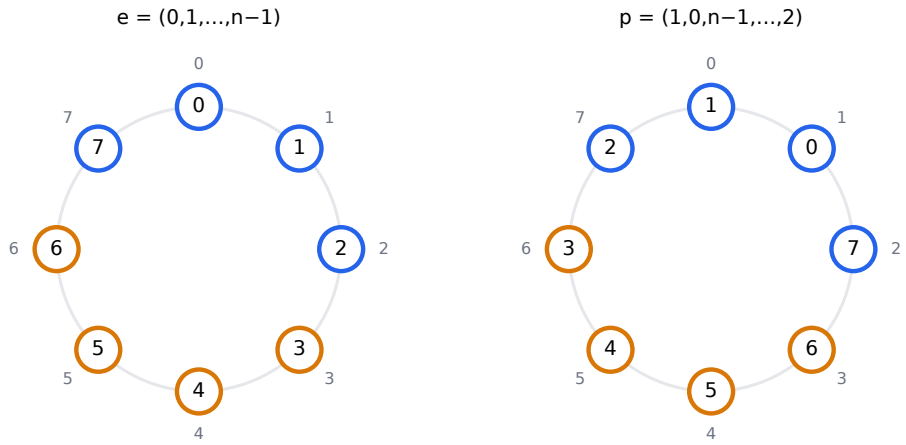


Figure 7: The two colour classes of a shortest word $e \to p^*$ for $n = 8$ (word found by breadth-first search; classes of 4 labels each). Grey numbers are places. Each class is an arc, and the order inside each arc is reversed.

transpositions, giving the reverse bound. Writing f for the fixed-point count and k for the nontrivial-cycle count gives $T = n - (f + k) = |\operatorname{supp} q| - k$. These unrestricted transpositions are not single LRX moves.

B.2 Affine length and the preservation of rotations

Set $F(i) = i + d_i$ for $0 \le i < n$ and extend by $F(i+n) = F(i) + n$. The lift congruence makes the residues $F(i)$ mod n distinct, hence F is a bijection of $\mathbb{Z}$. Balance gives $\sum_{i=0}^{n-1} F(i) = n(n-1)/2$. Thus F is in the affine symmetric group of type $\widetilde{A}_{n-1}$. We use its standard length formula [25, Proposition 3.1, pp. 6–7]:

$$C(F) = \sum_{0 \le i < j < n} \left| \left\lfloor \frac{F(j) - F(i)}{n} \right\rfloor \right|.$$

This is the minimum number of periodic adjacent exchanges realizing F. The source uses $1, \dots, n$; replacing F by $i \mapsto F(i-1) + 1$ gives exactly our convention. The periodic adjacent transpositions form a Coxeter system [25, Proposition 3.3, p. 8]. For an LRX word with this lifted endpoint, the deletion property [24, Proposition 1.4.7] allows deletion of pairs of X letters until exactly $C(F)$ remain, preserving every L/R letter and the lifted endpoint. Indeed, record each X as the periodic adjacent transposition at the current physical cursor; deleting X does not change that cursor sequence. This reduction determines the exchange count, but a separate construction is needed to bound the rotations.

There is also a trajectory description of deletion. If two lifted trajectories cross twice, delete the two crossings and exchange their names between those crossings, doing so for every periodic copy. The tokens exchanged have distinct residues, so their own periodic copies are never identified. The endpoint and cursor walk are unchanged. Iteration terminates, and each remaining trajectory pair crosses once precisely when its endpoint order is inverted. Thus exactly $C(F)$ exchanges remain.

B.3 The median estimate used for rotations

The fact that a median minimizes $\sum_i |x_i - t|$ is classical; see [34, p. 38]. Here $x_i = 2i$, $0 \le i < j$. For $j = 2k$ choose the median $t = 2k - 1$ and sum the odd numbers on either side: the minimum is $2(1 + 3 + \cdots + (2k-1)) = 2k^2$. For $j = 2k + 1$ choose the median $t = 2k$; the minimum is

$2(2+4+\cdots+2k) = 2k(k+1)$. In either case it is $\lfloor j^2/2 \rfloor$. This numerical estimate is separate from the proof that the projected path actually visits an external cursor in Appendix A.

C Majorization notation and the lift calculation

We use ordinary majorization as defined in [29, Section 3]. Let $d_{[1]} \geq \cdots \geq d_{[n]}$ be the sorted coordinates and $\rho_i = (n-1)/2 - i$, $0 \leq i < n$. The largest sum over k coordinates of d is $\sum_{r=1}^{k} d_{[r]}$, whereas the sum of the largest k coordinates of ρ is $k(n-k)/2$. Thus balance and the subset inequalities in Section 5 mean exactly $d \prec \rho$. The definition in the cited source uses nonnegative vectors; add $(n-1)/2$ to every coordinate of both vectors. The singleton and complementary-subset bounds ensure nonnegativity. Rado's theorem [29, Theorem 7] identifies the resulting condition with membership in the convex hull of the permutations of $(n-1, n-2, \ldots, 0)$.

The lift-specific energy expansion is

$$\begin{aligned}\mathcal{E}(d') - \mathcal{E}(d) &= 2k\sum_i d_i - 2n\sum_{i\in S} d_i + nk^2 - 2nk^2 + n^2k \\ &= n\Big(k(n-k) - 2\sum_{i\in S} d_i\Big).\end{aligned}$$

It uses $d'_i = d_i + k - n\mathbf{1}_{i\in S}$, $|S| = k$ and $\sum_i d_i = 0$. The identification with ordinary majorization is standard; verifying that this particular phase shift remains a lift and that energy minimization gives the inequalities is the application to our construction, retained in Section 5.

D Supplementary proofs for standard results

D.1 A rank proof of the Bruhat transfer

This appendix retains the proof of the standard lifting property [24, Theorem 2.1.5, Proposition 2.2.7] in the notation of Lemma 5.9.

Proof. Suppose q has a descent at position i and $p \preceq q$. Sort that pair in both maps, skipping the exchange in p if it has no descent. Call the results p', q'. Only rank row i can change. For a threshold between $q(i+1)$ and $q(i)$ put $A = \mathcal{R}_p(i-1,t)$, $B = \mathcal{R}_q(i-1,t)$, $x = \mathbf{1}_{\{p(i)>t\}}$, and $y = \mathbf{1}_{\{p(i+1)>t\}}$. Then $A \leq B$ and $A + x + y \leq B + 1$. The new row entries are $A + \min(x,y)$ for p' and B for q'. If $A < B$, $A + \min(x,y) \leq A + 1 \leq B$; if $A = B$, use $x + y \leq 1$. Outside this threshold range the row for q is unchanged and the row for p can only decrease. Hence $p' \preceq q'$. Induction carries p to the identity along with q. Every performed exchange is a descent, so exactly $C(F)$ are performed. Finally $p \preceq \mathrm{id}$ forces $p = \mathrm{id}$: the identity has the smallest possible ranks $\max(0, i-t)$, and the rank numbers determine the permutation. □

D.2 Shallow involutions and noncrossing chords

The classification used in Lemma 5.11 is [28, Corollary 25]. Here is the retained direct calculation. For an involution σ with h nontrivial pairs, let J count unordered pairs of chords whose four endpoints alternate in the linear order. Then

$$\Delta(\sigma) - I(\sigma) - h = 2J.$$

A single pair (a,b) contributes $2(b-a)$ to Δ and $2(b-a)-1$ to I, hence one to $\Delta - I$. Contributions between two pairs are computed by their four endpoints. Separate pairs $a < b < c < d$ paired as $(a,b),(c,d)$ and nested pairs $(a,d),(b,c)$ give no correction; alternating pairs $(a,c),(b,d)$ reduce the sum of the individual inversion counts by two. Inserting a fixed position adds the same $2w$ to Δ and I, where w is the number of pairs crossing its cut. Thus the formula also includes fixed points. Since $\mathrm{rk}(\sigma) = h$, shallowness is equivalent to $J = 0$.

E Complete C++17 certificate

The following is the complete program used in Section 6. It requires a C++17 compiler and reads no tables or external input data. Save the listing as `upper_bound.cpp` and run it with assertions enabled, as described in Section 7.

```
// Self-contained necessary-condition certificate for LRX.
// Usage: ./certificate [1|2]; no input data or external libraries.
#include <algorithm>
#include <cassert>
#include <cstdint>
#include <iostream>
#include <map>
#include <tuple>
#include <sstream>
#include <set>
#include <string>
using I = std::int64_t;

I floor_sqrt(I z) {
  if (z < 0) return -1;
  I lo=0, hi=z+1;
  while(lo+1<hi) { I x=lo+(hi-lo)/2; if(x*x<=z) lo=x; else hi=x; }
  return lo;
}

I ceil_sqrt(I z) {
  if (z <= 0) return 0;
  I lo=0, hi=z;
  while (lo<hi) { I x=lo+(hi-lo)/2; if (x*x>=z) hi=x; else lo=x+1; }
  return lo;
}


#include <array>
#include <vector>
#include <numeric>
#include <functional>
#include <cstdlib>
struct Local {
  std::vector<int> F,d; I f=0,inv=0,D=0,Z=0,S=0,c=0,U=0;
};
std::vector<Local> catalogue[6];
bool describe(const std::vector<int>& F,Local& z) {
  int n=F.size();z.F=F;z.d.resize(n);
  std::vector<int> id(n,-1),seen(n),load(n);
  std::vector<std::vector<int>> cycles,loads;
  for(int i=0;i<n;i++){
    z.d[i]=F[i]-i;z.D+=std::abs(z.d[i]);z.f+=F[i]==i;
    for(int j=i+1;j<n;j++)z.inv+=F[i]>F[j];
    for(int e=std::min(i,F[i]);e<std::max(i,F[i]);e++)load[e]++;
  }
  for(int e=0;e<n-1;e++)if(!load[e])return false;
  for(int i=0;i<n;i++)if(!seen[i]){
    std::vector<int> c;int j=i;do{seen[j]=1;c.push_back(j);j=F[j];}while(j!=i);
    if(c.size()>1){int k=cycles.size();for(int v:c)id[v]=k;cycles.push_back(c);}
  }
  int k=cycles.size();std::vector<std::vector<int>> reach(k,std::vector<int>(k));
  for(int j=0;j<k;j++){
    reach[j][j]=1;std::vector<int> a(n);
    for(int i:cycles[j]){
      for(int e=std::min(i,F[i]);e<std::max(i,F[i]);e++)a[e]++;
      for(int v=std::min(i,F[i])+1;v<std::max(i,F[i]);v++)
        if(id[v]>=0)reach[j][id[v]]=1;
    }
    loads.push_back(a);
  }
  for(int h=0;h<k;h++)for(int i=0;i<k;i++)for(int j=0;j<k;j++)
    reach[i][j]|=reach[i][h]&&reach[h][j];
  std::vector<int> rootLoad(n);int roots=0,qinside=0;
  for(int i=0;i<k;i++){
    bool root=std::all_of(reach[i].begin(),reach[i].end(),[](int v){return v;});
    if(root){roots++;for(int e=0;e<n;e++)rootLoad[e]+=loads[i][e];}
    else {qinside++;z.S+=*std::max_element(loads[i].begin(),loads[i].end())/2;}
```

```
  }
  assert(roots>0);
  z.U=*std::max_element(rootLoad.begin(),rootLoad.end())/2;
  z.Z=z.D-z.inv-n+z.f+k;z.c=z.Z-roots+z.S-qinside;
  assert(z.Z>=0&&z.Z%2==0&&z.c>=-1);return true;
}
void make_catalogue(){
  for(int n=2;n<=5;n++){std::vector<int> p(n);std::iota(p.begin(),p.end(),0);
    do{Local z;if(describe(p,z))catalogue[n].push_back(z);}
    while(std::next_permutation(p.begin(),p.end()));
  }
}
bool tail_ok(std::vector<I> values,I x,I p,I eps){
  std::sort(values.begin(),values.end());I sum=0;
  for(I r=1;r<=(I)values.size();r++){sum+=values[r-1];if((x+r)*(x+r+p)>eps+2*sum)return false;}
  return true;
}
bool signs_ok(const std::vector<const Local*>& parts,I f,I p,I eps,int extra=0){
  std::vector<I> pos,neg;
  for(const auto* z:parts)for(int d:z->d){if(d>0)pos.push_back(d);if(d<0)neg.push_back(-d);}
  if(extra>0)pos.push_back(extra);if(extra<0)neg.push_back(-extra);
  // x(x+p)<=eps implies -eps-2<=x<=eps+2 for p=0,1.
  for(I xp=-eps-2;xp<=eps+2;xp++){
    I xm=f-p-xp;if(xp*(xp+p)>eps||xm*(xm+p)>eps)continue;
    if(tail_ok(pos,xp,p,eps)&&tail_ok(neg,xm,p,eps))return true;
  }
  return false;
}
I cyclic_distance(I n,I a,I b){I d=std::abs(a-b);return std::min(d,n-d);}
I three_port_max(I n){
  I ans=0;
  for(I peak=0;peak<n;peak++)for(I s=0;s<n;s++)for(I t=0;t<n;t++){
    std::array<I,3> ports{0,2,peak};std::sort(ports.begin(),ports.end());I best=1000000;
    do{I value=cyclic_distance(n,s,ports[0])+cyclic_distance(n,ports[0],ports[1])
      +cyclic_distance(n,ports[1],ports[2])+cyclic_distance(n,ports[2],t);
      best=std::min(best,value);
    }while(std::next_permutation(ports.begin(),ports.end()));
    ans=std::max(ans,best);
  }return ans;
}
I attempts=0,geometry=0,creditCuts=0,bitonicCuts=0,tourCuts=0,survivors=0;
std::map<I,I> tourCache;
void refine(const std::string& row,I threshold){
  std::string clean=row;std::replace(clean.begin(),clean.end(),',',' ');std::istringstream in(clean);
  I t,m,p,f,eps,a,U,v,R,Told;in>>t>>m>>p>>f>>eps>>a>>U>>v>>R>>Told;assert(in);
  I n=2*m+p,L=n-a-v,x=L-R,B=a+t-(a>0),A=2*eps+p-f;
  assert(v-2*(t-1)<=5); // Verified for every generated scalar row, not assumed.
  std::vector<const Local*> parts;
  std::function<void(I,I)> go=[&](I left,I mass){
    if(left){
      for(I size=2;size<=std::min<I>(5,mass-2*(left-1));size++)
        for(const auto& z:catalogue[size]){parts.push_back(&z);go(left-1,mass-size);parts.pop_back();}
      return;
    }
    if(mass)return;attempts++;
    I fe=0,Ie=0,De=0,Se=0,Ce=0,Ue=0;
    for(const auto* z:parts){fe+=z->f;Ie+=z->inv;De+=z->D;Se+=z->S;Ce+=z->c;Ue=std::max(Ue,z->U);}
    I fb=f-a-fe;if(fb<0||Ue>U)return;
    if(Ue<U&&R<2*U)return;
    I cb=2*B-U-Ce-A-1-threshold;if(cb < -1)return;
    I T=2*B+m-U-2-eps-threshold-((v+a)*(v+a-1)/2-Ie)-Se;
    if(T<0||x>2*T+fb)return;
    I cap=m+p-2,qq=x/cap,rr=x%cap,S2=qq*cap*cap+rr*rr;
    I energy=std::min<I>(2*(S2/4),2*T*(T+(fb>0)));
    if(m*(m+p)-eps>2*U*(L-U)+De+energy)return;
    if((x+v+a)*(x+v+a)>4*eps+p+4*(De+energy))return;
    if(!signs_ok(parts,f,p,eps))return;
    geometry++;
    bool noUnit=!signs_ok(parts,f,p,eps,1)&&!signs_ok(parts,f,p,eps,-1);
    bool noTwo=!signs_ok(parts,f,p,eps,2)&&!signs_ok(parts,f,p,eps,-2);
    // Lemma: fixed-point-free, unit-free component has c>=0;
    // c=0 implies its root has k=Z=2 and no internal gaps.
    if(!fb&&noUnit&&(cb<0||(cb==0&&x>0))){creditCuts++;return;}
    // Lemma: if in addition no arrows of length 2 occur, nonempty
    // gaps imply c>=2. Proof uses Z>=2 per gap and bitonic Z>=4.
```

```
    if(!fb&&noUnit&&noTwo&&x>0&&cb<=1){bitonicCuts++;return;}
    // Exact exceptional route: outside two adjacent unit pairs,
    // core k=Z=2, U=4, no internal gaps, m<=10.
    if(!fb&&noUnit&&cb==0&&x==0&&t==2&&p==0&&f==0&&eps==0&&a==0&&U==4
       &&parts[0]->F.size()==2&&parts[1]->F.size()==2&&m<=10){
      if(!tourCache.count(n))tourCache[n]=three_port_max(n);
      assert(tourCache[n]<=n);assert(m-10<=0);tourCuts++;return;
    }
    survivors++;std::cout<<"SURVIVOR "<<row<<" cb="<<cb<<" fb="<<fb<<" x="<<x
      <<" noUnit="<<noUnit<<" noTwo="<<noTwo<<"\n";
  };
  go(t,v);
}


#include <algorithm>
#include <array>
#include <cassert>
#include <cstdint>
#include <iostream>
#include <numeric>
#include <vector>
using namespace std;
int fac[11];
int rank_perm(const array<int,10>& a,int n){
 int z=0;for(int i=0;i<n;i++){int c=0;for(int j=i+1;j<n;j++)c+=a[j]<a[i];z+=c*fac[n-1-i];}return z;
}
array<int,10> unrank(int z,int n){
 array<int,10>a{},left{};iota(left.begin(),left.end(),0);
 for(int i=0;i<n;i++){int k=z/fac[n-1-i];z%=fac[n-1-i];a[i]=left[k];for(int
 ↪  j=k;j<n-i-1;j++)left[j]=left[j+1];}return a;
}
void check_small_sizes(){
 fac[0]=1;for(int n=1;n<=10;n++)fac[n]=n*fac[n-1];
 for(int n=4;n<=10;n++){
  vector<int16_t>d(fac[n],-1);vector<int>q;q.reserve(fac[n]);q.push_back(0);d[0]=0;int diameter=0;
  for(size_t at=0;at<q.size();at++){
   int label=q[at];auto a=unrank(label,n);int next=d[label]+1;
   auto add=[&](const array<int,10>& b){int
   ↪  z=rank_perm(b,n);if(d[z]<0){d[z]=next;q.push_back(z);diameter=max(diameter,next);}};
   auto b=a;rotate(b.begin(),b.begin()+1,b.begin()+n);add(b);
   b=a;rotate(b.begin(),b.begin()+n-1,b.begin()+n);add(b);
   b=a;swap(b[0],b[1]);add(b);
  }
  assert(q.size()==(size_t)fac[n]);assert(diameter<=n*(n-1)/2);
  cout<<"BFS n="<<n<<" states="<<q.size()<<" diameter="<<diameter<<"\n"<<flush;
 }
}

int main(int argc, char** argv) {
  check_small_sizes();
  I threshold=argc>1?std::atoll(argv[1]):1;assert(threshold==1||threshold==2);
  std::map<std::string,std::tuple<I,I,I>> groups;
  std::set<std::string> rows;
  bool detail = false;
  I total=0, accepted=0;
  for (I t=2;t<=12;++t) {
    I checked=0, found=0;
    for (I p=0;p<=1;++p) for (I f=0;f<=t+5;++f)
    for (I eps=0;eps<=3*t+6;++eps) {
      if (f*f>4*eps+p || 2*eps+p<f) continue;
      I A=2*eps+p-f;
      for (I a=0;a<=f;++a) for (I U=1;U<=3*t+1;++U) {
        I B=a+t-(a>0), peak=2*B+t-U-A;
        if (peak<threshold) continue;
        I m0=std::max(2*t+a+1,t>=3?t*t*t-3*t*t+3*t+1:0);
        for (I m=m0;m<=1379;++m) {
          if ((m*(m+p)-eps)%2) continue;
          I n=2*m+p, cap=m+p-2;
          I v0=std::max(2*t,(2*f+t*t+2*t-eps+1)/2);
          I vmax=std::min(m-a-1,(2*B+m-U-2-threshold-eps+2*t*(t-1)-a*(a-1)/2)/(2*(t-1)+a));
          for (I v=v0;v<=vmax;++v) {
            I L=n-a-v;
            if (2*U>L) continue;
            I M=2*(t-1)*(v-t)+a*v+a*(a-1)/2;
```

```
            I cost=2*B+m-U-2-threshold-M-eps;
            if (cost<0) continue;
            I credit=4*cost*cost+2*cost;
            I ext=(v-2*t+2)*(v-2*t+2)+4*(t-1);
            I extEnergy=ext-(v%2);
            I need=2*(m*(m+p)-eps)-4*U*(L-U);
            I fix=(f>a?1:0);
            I energy=4*cost*(cost+fix);
            if(need>extEnergy+energy)continue;
            // x=L-R is the total mass of the big core's internal gaps.
            // Each piece of the square envelope is increasing in x;
            // the density expression on that piece is concave in x.
            for (I branch=0;branch<=1;++branch) {
              I lo,hi;
              if (branch==0) {
                lo=ceil_sqrt(need-ext); hi=std::min(cap,L-4);
              } else {
                lo=cap+ceil_sqrt(need-ext-cap*cap); hi=L-4;
              }
              lo=std::max(lo,need-ext-credit);
              hi=std::min(hi,floor_sqrt(2*A-2*v+1-p+2*(credit+ext))-(v+a-1));
              hi=std::min(hi,floor_sqrt(4*eps+p+2*energy+2*extEnergy)-v-a);
              hi=std::min(hi,2*cost+f-a);
              if (lo>hi) continue;
              for (I x=lo;x<=hi;++x) {
                I sq=ext+(branch==0?x*x:cap*cap+(x-cap)*(x-cap));
                assert(sq>=need);
                assert(ext+x+credit>=need);
                assert((x+v+a-1)*(x+v+a-1)+2*v-1+p-2*(credit+ext)<=2*A);
                I V=x+v;
                I density=V*V-2*sq+2*(V-1)*a+a*a+p;
                ++checked;
                if (density<=2*A) {
                  ++found;
                  std::ostringstream key, row;
                  key<<t<<","<<p<<","<<f<<","<<eps<<","<<a<<","<<U;
                  row<<t<<","<<m<<","<<p<<","<<f<<","<<eps<<","<<a<<","<<U<<","<<v<<","<<(L-x)<<","<<cost;
                  if(rows.insert(row.str()).second) {
                    auto it=groups.find(key.str());
                    if(it==groups.end()) groups[key.str()]={1,m,m};
                    else { auto &[cnt,mn,mx]=it->second; ++cnt; mn=std::min(mn,m);mx=std::max(mx,m); }
                  }
                }
              }
            }
          }
        }
      }
    }
    std::cout<<t<<" "<<checked<<" "<<found<<"\n"<<std::flush;
    total+=checked;accepted+=found;
  }
  std::cout<<"TOTAL "<<total<<" "<<accepted<<"\n";

  std::cout<<"UNIQUE "<<rows.size()<<" GROUPS "<<groups.size()<<"\n";
  for(const auto &[key,z]:groups) {auto [cnt,mn,mx]=z;std::cout<<"GROUP "<<key<<" "<<cnt<<" "<<mn<<"
  ↪  "<<mx<<"\n";}
  if(detail)for(const auto &row:rows)std::cout<<"ROW "<<row<<"\n";
  make_catalogue();for(const auto& row:rows)refine(row,threshold);
  std::cout<<"REFINE attempts="<<attempts<<" geometry="<<geometry<<" creditCuts="<<creditCuts
    <<" bitonicCuts="<<bitonicCuts<<" tourCuts="<<tourCuts<<" survivors="<<survivors<<"\n";
  for(const auto& [n,cost]:tourCache)std::cout<<"THREE_PORTS "<<n<<" "<<cost<<"\n";
  assert(survivors==0);
}
```

Reference output

For argument `1`, the complete standard output is:

```
BFS n=4 states=24 diameter=6
BFS n=5 states=120 diameter=10
BFS n=6 states=720 diameter=15
BFS n=7 states=5040 diameter=21
BFS n=8 states=40320 diameter=28
```

```
BFS n=9 states=362880 diameter=36
BFS n=10 states=3628800 diameter=45
2 12245 867
3 7 7
4 0 0
5 0 0
6 0 0
7 0 0
8 0 0
9 0 0
10 0 0
11 0 0
12 0 0
TOTAL 12252 874
UNIQUE 757 GROUPS 14
GROUP 2,0,0,0,0,3 6 10 14
GROUP 2,0,0,0,0,4 56 8 26
GROUP 2,0,0,0,0,5 193 8 36
GROUP 2,0,0,1,0,3 7 11 15
GROUP 2,0,1,1,0,4 15 15 23
GROUP 2,0,2,1,0,4 2 21 21
GROUP 2,0,2,1,0,5 33 19 31
GROUP 2,1,0,0,0,3 21 10 16
GROUP 2,1,0,0,0,4 132 7 27
GROUP 2,1,1,0,0,3 3 14 15
GROUP 2,1,1,0,0,4 53 14 24
GROUP 2,1,1,0,0,5 230 10 36
GROUP 2,1,1,0,1,5 1 29 29
GROUP 3,0,0,0,0,8 5 48 50
REFINE attempts=13009 geometry=691 creditCuts=682 bitonicCuts=7 tourCuts=2 survivors=0
THREE_PORTS 16 16
THREE_PORTS 20 20
```

References

[1] L. Babai and Á. Seress. On the diameter of Cayley graphs of the symmetric group. *Journal of Combinatorial Theory, Series A*, 49(1):175–179, 1988. doi:10.1016/0097-3165(88)90033-7.

[2] L. Babai and Á. Seress. On the diameter of permutation groups. *European Journal of Combinatorics*, 13(4):231–243, 1992. doi:10.1016/S0195-6698(05)80029-0.

[3] L. Babai, R. Beals, and Á. Seress. On the diameter of the symmetric group: polynomial bounds. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, pages 1108–1112. SIAM, 2004.

[4] E. Breuillard, B. Green, and T. Tao. Approximate subgroups of linear groups. *Geometric and Functional Analysis*, 21(4):774–819, 2011. doi:10.1007/s00039-011-0122-y.

[5] H. A. Helfgott and Á. Seress. On the diameter of permutation groups. *Annals of Mathematics*, 179(2):611–658, 2014. doi:10.4007/annals.2014.179.2.4.

[6] A. Chervov, D. Fedoriaka, E. Konstantinova, A. Naumov, et al. CayleyPy Growth: Efficient growth computations and hundreds of new conjectures on Cayley graphs (brief version). arXiv:2509.19162v2 [math.CO], 2026. First submitted in 2025; version 2, 1 April 2026; Conjecture 11. URL: https://arxiv.org/abs/2509.19162v2.

[7] Alexander Chervov, Kirill Khoruzhii, Nikita Bukhal, Jalal Naghiyev, Vladislav Zamkovoy, Ivan Koltsov, Lyudmila Cheldieva, Arsenii Sychev, Arsenii Lenin, Mark Obozov, Egor Urvanov, and Alexey M. Romanov. A machine learning approach that beats rubik's cubes. In *Advances in Neural Information Processing Systems*, volume 38, pages 191182–191204, 2025. doi:10.52202/085713-5741.

[8] A. Chervov, F. Levkovich-Maslyuk, A. Smolensky, F. Khafizov, I. Kiselev, D. Melnikov, I. Koltsov, S. Kudashev, D. Shiltsov, M. Obozov, S. Krymskii, V. Kirova, E. V. Konstantinova, A. Soibelman, S. Galkin, L. Grunwald, A. Kotov, A. Alexandrov, S. Lytkin, D. Fedoriaka, A. Chevychelov, Z. Kogan, A. Natyrova, L. Cheldieva, O. Nikitina, S. Fironov, A. Vakhrushev, A. Lukyanenko, V. Ilin, D. Gorodkov, N. Bogachev, I. Gaiur, M. Zaitsev, F. Petrov, L. Petrov, T. Gaintseva, A. Gavrilova, M. N. Smirnov, N. Kalinin, A. Khan, K. Jung, H. Mousset, H. Isambert, and O. Debeaupuis. CayleyPy-4: AI-holography. towards analogs of holographic string dualities for AI tasks. 2026. URL: https://arxiv.org/abs/2603.22195, arXiv:2603.22195.

[9] Michael R. Douglas and Kit Fraser-Taliente. Diffusion models for cayley graphs. 2025. arXiv:2503.05558.

[10] Yang-Hui He and Shing-Tung Yau. Graph laplacians, riemannian manifolds and their machine-learning. *Mathematics, Computation and Geometry of Data*, 2(1):1–48, 2022. arXiv:2006.16619, doi:10.4310/MCGD.2022.v2.n1.a1.

[11] Stephen McAleer, Forest Agostinelli, Alexander Shmakov, and Pierre Baldi. Solving the rubik's cube with approximate policy iteration. In *International Conference on Learning Representations*, 2019.

[12] Forest Agostinelli, Stephen McAleer, Alexander Shmakov, and Pierre Baldi. Solving the rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8):356–363, 2019. doi:10.1038/s42256-019-0070-z.

[13] Kyo Takano. Self-supervision is all you need for solving rubik's cube. *Transactions on Machine Learning Research*, 2023. arXiv:2106.03157.

[14] Colin G. Johnson. Solving the rubik's cube with learned guidance functions. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2082–2089, 2018. doi:10.1109/SSCI.2018.8628626.

[15] Rashid Barket, Enrico Grimaldi, Yacoub Hendi, Edward Hirst, Adam Onus, and Harmeet Singh. Learning the graphical nature of symmetries. 2026. arXiv:2607.12026.

[16] OEIS Foundation Inc. A186783: Diameter of the symmetric group S_n when generated by the transposition $(1, 2)$ and both left and right rotations by $(1, 2, \ldots, n)$. The On-Line Encyclopedia of Integer Sequences, 2026. Entry created by Tony Bartoletti in 2011; accessed 4 November 2026. URL: https://oeis.org/A186783.

[17] A. Chervov, M. Obozov, A. Soibelman, S. Lytkin, et al. CayleyPy RL: Pathfinding and reinforcement learning on Cayley graphs. arXiv:2502.18663v3 [cs.LG], 2026. First submitted in 2025; version 3, 15 May 2026; Sections 4.1–4.3. URL: https://arxiv.org/abs/2502.18663v3.

[18] Grigorii Antiufeev. A lower bound for the diameter of cayley graph of the symmetric group S_n generated by (12), $(12\ldots n)$, $(1n\ldots 2)$. arXiv:2601.08715v3 [math.CO], 2026. Version 3, 4 June 2026; Theorem 2. URL: https://arxiv.org/abs/2601.08715v3.

[19] V. M. Glushkov. Completeness of a system of operations in digital computers. *Cybernetics*, 4(2):1–4, 1968. Translated from Kibernetika 4(2) (1968), 1–5. doi:10.1007/BF01073731.

[20] B. A. Pogorelov. A problem of V. M. Glushkov. *Cybernetics*, 9:895–899, 1973. Translated from Kibernetika (1973), no. 5, 141–143. doi:10.1007/BF01069732.

[21] M. M. Glukhov and A. Yu. Zubov. On the lengths of symmetric and alternating permutation groups in various generating systems (a survey). In *Matematicheskie Voprosy Kibernetiki*, volume 8, pages 5–32. Nauka, Moscow, 1999. In Russian. URL: https://keldysh.ru/papers/1999/mvk/mvk1999_5.pdf.

[22] OEIS Foundation Inc. A039745: Diameter of symmetric group S_n when generated by $(1, 2)$ and $(1, 2, 3, \ldots, n)$. The On-Line Encyclopedia of Integer Sequences, 2026. Entry by D. desJardins; accessed 5 November 2026. URL: https://oeis.org/A039745.

[23] Sergei Uskov, Alexander Chervov, Fedor Levkovich-Maslyuk, Ivan Litvak, Andrei Smolensky, and Alexander Soibelman. Exact diameter of the symmetric group with LX generators. Hexagon preprint hexagon:2611.00039v1, 2026. Version 1, 3 November 2026. URL: https://hexagonmath.org/2611.00039v1.

[24] Anders Björner and Francesco Brenti. *Combinatorics of Coxeter Groups*, volume 231 of *Graduate Texts in Mathematics*. Springer, 2005. Proposition 1.4.7, Corollary 1.4.8, Proposition 1.5.2, Theorems 2.1.5 and 2.2.2, Corollary 2.2.5, Propositions 2.2.7 and 2.3.1. URL: https://link.springer.com/book/10.1007/3-540-27596-7, doi:10.1007/3-540-27596-7.

[25] Anders Björner and Francesco Brenti. Affine permutations of type A. *The Electronic Journal of Combinatorics*, 3(2):R18, 1996. Propositions 3.1 and 3.3, pp. 6–8. URL: https://www.combinatorics.org/ojs/index.php/eljc/article/view/v3i2r18, doi:10.37236/1276.

[26] Persi Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977. Theorem 2, pp. 264–266. URL: https://mathweb.ucsd.edu/~ronspubs/77_04_spearmans.pdf, doi:10.1111/j.2517-6161.1977.tb01624.x.

[27] T. Kyle Petersen and Bridget Eileen Tenner. The depth of a permutation. *Journal of Combinatorics*, 6(1–2):145–178, 2015. Theorem 1.1, Remark 1.2, Proposition 3.2; arXiv:1202.4765v3, 19 August 2014. URL: https://arxiv.org/abs/1202.4765v3, doi:10.4310/joc.2015.v6.n1.a9.

[28] Yosef Berman and Bridget Eileen Tenner. Pattern-functions, statistics, and shallow permutations. *The Electronic Journal of Combinatorics*, 29(4):P4.43, 2022. Definition 2, Corollaries 25 and 31; journal numbering. URL: https://www.combinatorics.org/ojs/index.php/eljc/article/view/v29i4p43, doi:10.37236/10858.

[29] Melvyn B. Nathanson. The Muirhead–Rado inequality, 1: Vector majorization and the permutohedron. arXiv:2109.01746v3 [math.CO], 2024. Version 3, 5 May 2024; Section 3 and Theorem 7. URL: https://arxiv.org/abs/2109.01746v3.

[30] Frank Harary and Jerald A. Kabell. Counting balanced signed graphs using marked graphs. *Proceedings of the Edinburgh Mathematical Society*, 24(2):99–104, 1981. Structure Theorem for Balance, Section 1, p. 99. doi:10.1017/S0013091500006398.

[31] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. Section 5.2, pp. 123–126: Balance Theorem and proof. URL: https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book-ch05.pdf.

[32] Andrea Guerrieri. In preparation.

[33] S. Uskov, A. Chervov, F. Levkovich-Maslyuk, I. Litvak, A. Smolensky, and A. Soibelman. Exact directed L/X diameter in Lean 4. https://github.com/snow-ghost/lrx-lean/releases/tag/v1.0.0, November 2026. Lean 4 formalization, version v1.0.0, commit 9e04062.

[34] Neil C. Schwertman, A. J. Gilks, and J. Cameron. A simple noncalculus proof that the median minimizes the sum of the absolute deviations. *The American Statistician*, 44(1):38–39, 1990. URL: https://tommasorigon.github.io/StatI/approfondimenti/Schwertman1990.pdf, doi:10.1080/00031305.1990.10475690.