ResearchArena-CayleyBench: RL/LLM Benchmark challenges which can advance mathematical research

Dmitrii Kovalenko, Dmytro Fedoriaka, Mark Obozov*, Elena V. Konstantinova, Anastasia Sheveleva, Alexander Soibelman, Igor Kiselev, Fedor Levkovich-Maslyuk, Dmitry Volovich, Nick Vilkin-Krom, Alim Bidzhiev, Artem Krasnyi, Mikhail Evseev, Artem Isakov, Michael Litvinov, Elizaveta Geraseva, Liliya Grunwald, Sergei Lytkin, Andrei Smolensky, Sergey Galkin, Eduard Koldunov, Anton Naumov, Ivan Koltsov, Stanislav Diner, Artem Chevychelov, Evelina Kudasheva, Arsenii Sychev, Zakhar Kogan, Altana Natyrova, Ruslan Grimov, Lidia Shishina, Lyudmila Cheldieva, Vladislav Zamkovoy, Dmitrii Kovalenko, Oleg Papulov, Kudashev Sergey, Dmitry Shiltsov, Rustem Turtayev, Olga Nikitina, Dariya Mamayeva, Nikolenko Sergei, Anton Titarenko, Antonina Dolgorukova, Anton Kostin, Alexey Kravchenko, Alexey N. Aparnev, Orianne Debeaupuis, Herve Isambert, Simo Alami Chehboune

Abstract

This is the third paper of the CayleyPy project applying artificial intelligence methods to problems in group theory. We announce the first public release of CayleyPy, an open-source Python library for computations with Cayley and Schreier (coset) graphs. Compared with state-of-the-art systems based on classical methods, such as GAP and Sage, CayleyPy handles significantly larger graphs and performs many orders of magnitude times faster.

Using CayleyPy we obtained about 200 new mathematical conjectures on Cayley and Schreier graphs which can be turned into an efficient benchmarks for both RL and LLM models. For many Cayley graphs of symmetric groups S_n we observe quasi-polynomial diameter formulas: a small set of quadratic or linear polynomials indexed by $n \mod s$ and conjecture that it is general phenomenon. We conjecture improved Babai-type bounds of the diameters by $\frac{1}{2}n^2 + 4n$ for undirected case, by $\frac{3}{4}n^2 + O(n)$ for directed cases, and by $\frac{1}{4}n^2 + O(n)$ for certain Schreier graphs, comparing to prior conjectural bounds of $O(n^2)$. For nilpotent groups we conjecture an improvement of J.S. Ellenberg's results on the diameter of the upper-triangular matrices over $/p$, presenting a phenomenon of linear dependence of the diameter with respect to p. Moreover, the growth for nilpotent groups is conjectured to follow Gaussian distributions, that is, to exhibit a central limit phenomenon similar to results of P. Diaconis for S_n.

Introduction

Context and motivation

Deep learning methods allowed to achieve significant progress in various fields of science and technology. Current momentum is characterized by the interest to apply these methods to mathematical tasks, with broader context to achieve ability of AI to produce verifiable and correct reasoning and ability to solve research tasks, not only in mathematics but also for coding and other fields, i.e. not to make mistake and be able to make research. One of the obstacles for these developments is lack of high quality data. Indeed, most of mathematics exists in the form of research papers and textbooks, which are not formalized enough to serve clear data for training LLM to solve research problems. In recent years several datasets appeared which are intended to mitigate that issue, e.g. [Saxton et al.(2019)Saxton, Grefenstette, Hill, and Kohli, He et al.(2025)He, Liang, Xu, and et al, Zheng, Han, and Polu(2021), Paster et al.(2023)Paster, Santos, Azerbayev, and Ba]. And there is huge progress in creating formalized proofs e.g. in Lean [The mathlib Community(2020), Carneiro(2018), Browning and Lutz(2021), Wieser and Song(2022), Loeffler and Stoll(2025), del Barco et al.(2025)del Barco, Infanti, Rivas, and Schwahn, Manthe(2025), Otte(2025)].

Fundamental problem. Nevertheless current amount of training data for research level mathematical tasks is far from the level where standard RL-based LLM training methods might lead to abilities of LLM to really solve wide range of mathematical research tasks. Research level mathematics is very diverse, and covering even part of its major fields by high quality data is hardly possible in near future within framework of standard approaches, which rely on manual human work in that or another way.

Present contributions. The contribution of the present paper is to make yet another step in resolving these issues, which is significantly different from all approaches considered before.

Moreover the proposed benchmarks serve several goals simultaneously : first to test ability of LLM to solve research level tasks; second to benchmark RL approaches on non-trivial, googol size environments; third any progress achieved on these tasks would advance open mathematical research problems.

And the other line of the contributions of the present paper is to present a tool, which serves as a key for all these developments - the first release of CayleyPy, an AI-based open-source Python library which can handle googol size graphs. Moreover, to present nearly 200 new mathematical conjectures obtained with it, and which can serve for both purposes: as a tool to check LLM and RL methods and vice versa LLM and RL might help to advance some of them.

In a nutshell the proposed benchmarks have very simple form: all of them are sorting problems, i.e. one needs to bring given input vectors to sorted state, the main difficulty that sorting is allowed to be performed only with specific permutations transformations. Each challenge differs by the set of the allowed transformations, mathematically called "generators of the group". The well-known examples of such kind of tasks are classical bubble-sort algorithm where allowed

*obozov.mark@innopolis.ru, Research Center of the Artificial Intelligence Institute Innopolis University

transformations are neighbor permutations $(i, i+1)$, and prefix (pancake) sorting where allowed transformations are flips of the first k-positions.

LLM approach. Such problems can be approached from LLM side - the task of LLM is to generate algorithms (Python code) which would be able to sort vectors using only the transformations specific to each particular dataset.

RL approach. And completely in another direction - from the point of classical RL approaches. To explain that framework - we need to observe that our tasks are exactly particular cases of the classical RL setup; indeed, there is the following vocabulary: states - permutations; actions - allowed transformations ("generators of group"), rewards (more precisely penalties) - always 1, unless you achieve the sorted state achieved where penalty is zero, cumulative reward - number of operations. So the ideal goal is to achieve sorted state using as minimal number of operations as possible. Which fits classical RL setup by the vocabulary above.

Mathematical significance of these kind of tasks is rather immediate, for most of them there are no known algorithms which can perform sorting, for some there are known algorithms, but they are sub-optimal, and better optimality is desirable. Some of these problems are open for more than 50 years, and any progress would be valuable. Moreover they represent a more general direction in mathematics - studying the Cayley graphs, fundamental objects in group theory which associated with any group and its generators. Our particular setup is related to permutation groups, and allowed transformations are precisely the generators of these groups. The sorting task becomes a particular case of the graph pathfinding task, by the main property of the Cayley graphs - paths on them from A to B, represent sequence of generators such that multiplication of these generators on A will give B. In our case "B" is the sorted state and thus one achieves sorting strictly by application of only allowed transformations ("generators"). The worst case performance of the algorithm becomes what is called "diameter" of the Cayley graph, which is just the length of the shortest path between the two most distant nodes. Diameters of the Cayley graphs is hard to estimate and subject of many fundamental research by leading mathematicians e.g. T. Tao and fundamental conjectures like L.Babai-like conjecture which in particular predicts that the diameter of the permutation groups Cayley graphs is $O(n^2)$. Any algorithm designed for specific allowed moves would give a constructive upper bound on the diameter of the corresponding Cayley graph, which in most cases would be a new research level result.

Detailed results.

1. We propose more than 10 benchmark datasets framed as Kaggle community challenges, to benchmark LLM abilities to solve research problems and on the other side to benchmark RL approaches to solve problems with very huge environment like googol size. Kaggle setup provides public visibility, clearly defined rules, gamified framework. Each dataset corresponds to different Cayley graph (i.e. choice of "allowed moves", group generators) . All them are framed as sorting problem, where sorting should be achieved only by using these allowed moves.
2. We provide mathematical conjectures and more open ended research problems for each dataset, such that advances in AI part would lead to progress in open mathematical conjectures, some of which are more than 50 years old and related to fundamental conjectures in the field like L. Babai conjecture and P. Diaconis conjecture.
3. We present the first release of CayleyPy - AI-based open source Python library to work with extremely huge graphs (googol size) with current main focus on the Cayley graphs of finite groups. It outperforms standard computer algebra systems like GAP/SAGE by many orders of magnitude on several tasks. In particular the approach of CayleyPy may serve as a baseline for RL sides of the proposed benchmark challenges, and any progress in benchmark task would allow to improve future releases of CayleyPy.
4. By means of CayleyPy we generated more around 200 hundred mathematical conjectures, and the progress in benchmark challenges which we propose, would all allow to advance these conjectures in particular. Some of these conjectures potentially open a new approach and look on such fundamental conjectures like Babai conjecture on the diameter of finite simple groups, which is still widely open, despite huge progress achieved by leading mathematicians like T.Tao.

Background and related studies

Caley and Schreier graphs Let us consider a finite (or possibly infinite) group G and its set of generators $S \subseteq G$, which can be chosen arbitrarily. The *Cayley graph* of the group G with respect to S is defined as a graph whose vertex set is G, and whose edges connect vertices g and gs for each $g \in G$ and $s \in S$.

The set S determines the structure of the Cayley graph. For example, if for each $s \in S$ we have $s^{-1} \in S$, then the resulting graph is undirected. Furthermore, the standard assumption $e \notin S$, where e is the identity element of G, guarantees that the Cayley graph contains no loops.

A more general construction, extending the notion of Cayley graphs, is given by *Schreier graphs*. Consider a group G and its subgroup $H \leq G$. Together with the set of generators $S \subseteq G$, we define a graph whose vertices are the right cosets $Hg = \{hg,\, h \in H\}$ for each $g \in G$. The set of edges is determined by the action of the generators S on these cosets: precisely, the pair $\{Hg, Hgs\}$ forms an edge for each $g \in G$ and $s \in S$. Thus, it is easy to see that when $H = \{e\}$, the Schreier graph coincides exactly with the Cayley graph.

From now on, let us fix one vertex v_0 of one of the graphs considered above, and let V denote its vertex set. The *distance* $d(v, u)$ between any two vertices v, u of a graph is defined as the minimum number of edges in a path connecting them. Using this distance, we define the ball of radius d around the vertex v_0 as follows

$$B(v_0, d) = \{v \in V \mid d(v_0, v) \leq d\}.$$

The number of vertices in the ball $B(v_0, d)$ is called the *growth function* of the graph with respect to the vertex v_0, and is denoted by

$$N(d) = |B(v_0, d)|.$$

There are enormous number of works on Cayley graphs, some monographs are: [Gromov(1993)], [Tao(2015)]. There are also various applications: in bioinformatics [Hannenhalli and Pevzner(1995), Hannenhalli and Pevzner(1999), Bulteau and Weller(2019)] for estimation of the evolutions distance ; processor interconnection networks [Akers and Krishnamurthy(1989), Cooperman, Finkelstein, and Sarawagi(1991), Heydemann(1997)],

Diameter estimation Another quantity derived from the graph distance is the *diameter* of the graph. It is defined as the greatest distance between any two vertices in the graph

$$\mathrm{diam} = \max_{v,u \in V} d(v, u).$$

Where distance on the graph is lenght of the shortest path. Estimating diameters of various Cayley and other graphs is typically a difficult problem, but this problem is getting a huge interest in a field of mathematics. It was proved to be NP-hard for general Cayley graphs [Even and Goldreich(1981)], it is very difficult to find for particular cases - it took 40 years to determine "God's number" of the standard Rubik's cube [Rokicki(2014)], and it still unknown for its higher versions.

Growth of the graph with respect to some selected node v is the sequence of integer numbers counting the number of nodes which are at distance k from v, where by distance length of the shortest path is meant, (as usually). Typically one starts from $k = 0$ and hence 1 the first term in growth sequence (because starting node v itself is the only at distance zero from itself), for $k = 1$ it equals to number of edges from v, and so on.

Fundamental problems of group theory Cayley graphs are fundamental in group theory [Gromov(1993)],[Tao(2015)], and have various applications: bioinformatics [Hannenhalli and Pevzner(1995), Hannenhalli and Pevzner(1999), Bulteau and Weller(2019)]; processor interconnection networks [Akers and Krishnamurthy(1989), Cooperman, Finkelstein, and Sarawagi(1991), Heydemann(1997)]; coding theory and cryptography [Hoory, Linial, and Wigderson(2006), Zémor(1994), Petit and Quisquater(2011)]; quantum computing [Ruiz(2024), Sarkar and Adhikari(2024), Dinur et al.(2023)Dinur, Hsieh, Lin, and Vidick, Acevedo, Roland, and Cerf(2006), Gromada(2022)], etc.

There are many open conjectures in the subject and making progress in their understanding is a fundamental challenge in the field. Two of these that are quite well-known, easy to formulate, wide open and most relevant to us are:

- **Babai-like conjecture:** for any choices of generators the diameter of S_n is $O(n^2)$ (see, e.g., [Helfgott and Seress(2014)], [Helfgott(2019)], [Helfgott, Seress, and Zuk(2015)]);
- **Diaconis conjecture [Diaconis(2013)]:** the mixing time for random walks is $O(n^3 \log n)$ (again for any choices of generators).

More generally the so-called "growth" (i.e. sizes of spheres of each radius $r \in N$) is important characteristic of a Cayley graph.

So having some elements in say permutation group S_n (or other group) one constructs a Cayley graph and there is a set of natural questions:

- What group is obtained ?
- Diameter ?
- Growth statistical characteristics: mean, mode, moments, what distribution is obtained ? (at least in the limit $n \to \infty$) ?
- Algorithm: is there an effective/polynomial algorithm which decomposes given element into product of generators (optimally/sub-optimally) ?
- Antipodes ("super-flips"): is there explicit description of the longest elements ?
- Can one explicitly describe the word-metric (i.e. number of generators in the decomposition of an element, i.e. length of the shortest path on a Cayley graph) ?
- Spectrum: what can be said about graph spectrum ?
- What is the mixing time ?

For vast majority of the Cayley graphs typically most of the questions are unresolved.

On the one hand there are theoretical limits which restricts hopes for complete solutions of the problems above. Finding the shortest paths on generic finite Cayley graphs is an NP-hard problem [Even and Goldreich(1981)] (even P-space complete [Jerrum(1985)]). And NP-complete is the case for many specific group families, such as $N \times N \times N$ Rubik's Cube groups [Demaine, Eisenstat, and Rudoy(2017)] and others [Bulteau, Fertin, and Rusu(2015)]. Determining the diameters for general groups is NP-hard (again [Even and Goldreich(1981)]).

On the other hand there are positive results for many generators and it is huge and active field of research to study questions similar to the above. For example, Coxeter's generators $(i, i+1)$ represents an extreme case when almost all questions have well-known and beautiful answers.

Decomposing elements into products of generators (path-finding on Cayley graphs)

Another relevant line of research is the algorithmic problem of decomposing an element of the group into a product of generators — in other words, solving Rubik's cube or other puzzles, or Cayley graph path-finding, or the sorting problem in computer science (all these formulations are equivalent). Some important milestones:

- The Schreier–Sims algorithm [Sims(1970)] can in principle work for arbitrary permutation groups. Its improved randomized version by Donald Knuth [Knuth(1991)] is implemented in GAP/SAGE. However, this algorithm is known to be impractical for large groups (e.g., of order 10^{40}), as the outputs "are usually exponentially long" [Fiat(1989)].
- It is NP-hard to **optimally** decompose elements for generic finite groups [Even and Goldreich(1981)], improved to P-space complete in [Jerrum(1985)]
- (1998-now) **Optimal** decomposition is NP-complete for many concrete families of generators like Rubik's cube [Demaine, Eisenstat, and Rudoy(2017)], pancakes [Bulteau, Fertin, and Rusu(2015)], etc.
- Optimal decomposition nevertheless can be achieved for groups of non extra-huge sizes: [Korf(1997)] proposed the general method of "pattern databases" and provided the first demonstration of the possibility to solve the $3 \times 3 \times 3$ Rubik's cube (4.3×10^{19} states) optimally. The solver was very slow, but it is currently improved to several cubes per second. A big challenge is achieving optimal solution for higher group sizes like 10^{30}–10^{40}; hopefully, it might be resolved by machine learning which is one of the CayleyPy project goals.
- Some examples of algorithms for particular generator families are known. [Bafna and Pevzner(1998)] presented a surprising breakthrough by showing that despite usual reversal sorting being NP-complete, for **signed** reversals they found an **optimal polynomial** algorithm. This made possible effective computations of evolutionary distance in biology and stimulated many further developments (survey: [Bulteau and Weller(2019)]). Classical algorithms include bubble sort – for Coxeter generators (optimal), pancake sorting algorithm (suboptimal) [Gates and Papadimitriou(1979)], Rubik's cube solvers (suboptimal), etc. [Larsen(2003)] proposed an algorithm for $SL_2(/p)$ with complexity $O(\log(p) \log(\log p))$, near optimal $O(\log p)$. Participants of the Kaggle Challenge Santa 2023 proposed methods which can effectively solve some puzzles with sizes up to 10^{1000} (like Rubik's cube $33 \times 33 \times 33$). They used a remarkable idea: first finding "small support" elements expressed via original generators, then using these new small support generators one can, for example, run beam search just with Hamming distance as guiding heuristics. However the generality of such an approach is unclear — it is unknown and not

even investigated by mathematical community, to the best our knowledge, for what permutation groups those "small support" elements can be effectively found. It is known that a restriction on the support of even a single generator of the form $\mathrm{supp} \leq 0.63n$ implies a polynomial bound on the diameter [Bamberg(2014)], see also A Seress's slides.

After the deep learning revolution it became natural to try deep learning methods on this problem. However, systematic investigations applying deep learning across diverse families of Cayley graphs appear to have been limited prior to our project. Nevertheless for the specific case of $3 \times 3 \times 3$ Rubik's cube there were two notable works which have demonstrated that deep learning methods can effectively solve it: the https://deepcube.igb.uci.edu/DeepCube series of papers [McAleer(2019), Agostinelli(2019), Khandelwal, Sheth, and Agostinelli(2024), Agostinelli(2024)], and later the Efficient Cube: [Takano(2021)]. Some others [Brunetto and Trunda(2017), Johnson(2021), Amrutha and Srinath(2022), Noever and Burdick(2021), Chasmai(2021), Bedaywi, Longtai, and Shaul(2023), Pan and Kondor(2021)] proposed several approaches, but did not achieve a solution. One noteworthy idea [Pan and Kondor(2021)] is combining neural networks with the representation theory of the symmetric group — a neural net predicts the coefficients of the non-abelian Fourier transform for the distance function. The rationale is to observed sparsity (bandlimitedness) of the Fourier transform of the common distance functions on S_n [Swan(2017)].

Results and proposal: LLM and RL benchmarks which can advance mathematical research open problems

The main goals of the present paper is to propose benchmark datasets which not only benchmark AI, but advances for which would lead to a progress in open problems in research level mathematics. Currently we propose more than 10 Kaggle challenges as benchmarks, but one can generate similar problems in rather unlimited manner. The second goal is to present CayleyPy - AI-based open source Python library which can work with googol size Cayley graphs. Which can provide a benchmark for RL counterpart of the challenges, by which we generated around 200 conjectures, which on the hand provide guidance what should be achieved on AI side, on the hand they can be advanced by AI methods.

For puzzle groups most of the states are created by simple rule k-th state to be solved is obtained by scrambling k-times the solved state for $k = 1...1000$. Thus first states are easy to solve, and may give user chance to start from accessible tasks. Moreover optimal solution would allow to get estimation of the mixing time for puzzle, which as an important question in mathematics. Because mixing time is roughly speaking the step k, when k-times scrambled state achieves mean diameter.

LLM Benchmarks Benchmarks are key instruments to measure capabilities of LLMs in different fields, especially in abilities to reason. Modern LLMs are able to generate working code on different programming languages, so we can measure their abilities in this field either. Unfortunately, most of the benchmarks became not difficult enough to the modern LLMs. Furthermore, while they are complicated, usually computational heavy benchmarks do not really bring value from their solutions. The proposed challenges above are non-trivial research problems in mathematics. The task of LLM is to generate algorithms (say Python code) which would solve the sorting tasks above. According to our preliminary tests current LLM are not at the level of solving such task at the level which would be help to advance research problems. In most cases they are not capable to generate and correct solutions at all, on in some simple cases able to find some solutions, but which are rather trivial and do allow to get new insights into the problems.

Mathematical contributions

The aim of the present paper is to make progress in direction of fundamental problems described above (i.e. understanding of various properties of Cayley graphs) with the help of the new tool which we are developing: AI-based Python open-source library CayleyPy which allows to make computational experiments orders of magnitude more effectively than standard computer algebra systems GAP/SAGE. We show that the pipelines which were introduced previously for some specific S_n sub-groups are scaling quite well for Cayley graph tasks Furthermore, due to the permutation-like structure of the most of the problems they can be formulated as a sorting problems, which are easy to formulate for LLM, and their solutions can be given by an algorithm or by a Python code, are easy to verify, so they can be used to test LLM's abilities to solve research problems. Meanwhile, our code for direct growth computation outperforms similar functions on the standard computer algebra system GAP/SAGE up to 1000 times both in speed and in maximum sizes of the graphs that it can handle.

- We generate around 200 conjectures on various properties of Cayley graphs, that is achieved by extensive computational experiments with around half hundred of Cayley graphs. The conjectures are summarized in tables 2,3. All these generators included into CayleyPy as named generators and results of hard computations available on github. More than 200 notebooks with computational experiments are publicly available on Kaggle platform, where it is easy to reproduce them.
- In particular we propose the following:
  - We conjecture that diameters of many S_n-Cayley graphs are quasi-polynomials (quadratic/linear) in n (i.e. several polynomials depending on n modulo some s) allowing to find them rather efficiently, which is surprising since it is NP-hard in general.
  - The improvement of the L.Babai-like conjecture for S_n - diameters are bounded by $n^2/2+4n$, by $3n^2/4+O(n)$ (directed cases), $n^2/4+O(n)$ for some Schreier graphs, comparing to prior $O(n^2)$ conjectural bounds. Moreover we present explicit families of generators for S_n which conjecturally provide largest (or near) diameters. They are related to involutions and follow rather simple pattern (""square-with-whiskers""). They were found by an extensive (partly exhaustive) search for $n <= 15$ of the generators with maximum diameter.
  - For nilpotent groups we conjecture improvement of J.S. Ellenberg's results on diameter of upper-triangular matrices over Z/p presenting phenomena of linear dependence of diameter on p. Moreover growth for nilpotent groups conjectured to follow Gaussian distributions (a central limit phenomena - similar to results of P.Diaconis for S_n).
  - We present a conjectural answer on the open question: diameter of the directed Cayley graph generated by left cyclic shift and transposition $(1, 2)$ is equal to $(3n^2 + 8n + 9)/4$ for odd n, else $(3n^2 - 8n + 12)/4$.
- To benchmark various methods of path-finding on Cayley graphs and LLMs we create 11 benchmark datasets in the form of Kaggle challenges, making benchmarking easy and public to community.

New conjectures and experimental results

We conducted extensive computations computing growth for large number of Cayley and certain Schreier coset graphs for up to $n \leq 15$ and $n \leq 42$ respectively. Obtained results and conjectures are summarized in the tables discussed below, which also include results known in the literature.

We analyzed not only diameters but other growth characteristics as a probability distribution - mean, mode, variance, skewness, kurtosis, tried to fit the distribution by some known like Gaussian or Gumbel, antipodes (longest elements, or super-flips), spectrum of the Cayley graph. In some cases we observe that growth by itself might have close analytical formula - given by Stirling numbers or related to Fibonacci numbers or e.g. coincide with some known sequence e.g. https://oeis.org/A367270OEIS-A367270 (”Growth/F-la” column of the table). For diameters in most (but not all) cases we able to fit by quasi-polynomials, for some cases, apparently, available data is not enough. But for mean diameters and other characteristics of growth there are not quasi-polynomials in general, for example literature contains results $n - log(n)$ for mean diameters, Nevertheless we expect that our numerical fits for the data provides approximations to the leading terms of these characteristics. They are obtained as fit on for small values of n and can be considered as conjectures for large values.

Notations used in the table:

1. ◆ - conjecture obtained by CayleyPy project, ◆ - proved by CayleyPy
2. $+$ information is known/conjectured - can be found in the main text (too big to fit into table)
3. $*$ - conjecture from the literature
4. ? - no information, neither in literature, nor our experiments suggest clear pattern
5. notations like $1|2$ indicates 1 for even n and 2 for odd (or vice versa)
6. notations $+I$ - some quasi-polynomial typically of zero degree
7. ”Group” information on generated group, if just $+$ information is known, but not fits into the table
8. ”Growth/PDF” - what continuous distribution fits growth for large n (Gaussian, or Gumbel, etc)
9. ”Growth/F-la” - explicit formulas for the growth
10. ”Antipode” - information on longest elements - i.e. if there explicit description, if the number is known (and simple to fit into table) we indicate it, or simple write $+$
11. ”Algorithm” - indicates is there known algorithm to decompose element into product of these generators, the upper-script O indicates that optimal algorithm is known, notations like $NP/2$ means that optimal decomposition was proved to be NP-hard, but there polynomial approximations by factor of 2.
12. ”Metric” - is there explicit expression for the word metric for given generators, for example for Coxeter generators it is a number of inversions
13. ”Spectrum” - information on spectrum, ”Int” integer spectrum, ”Wig” - approaches Wigner semi-circle law for large n, ”Uni” - almost uniform

Benchmark: LLMs on Puzzle Solving

Setup

LLMs were given descriptions of different puzzles and generators. They were prompted to generate an optimal and polynomial (if possible) algorithm that will solve the puzzle. Then, their algorithms were validated on test sets from prepared Kaggle challenges and rated in terms of complexity and optimality.

Results

Puzzle	**GPT-5**	**Gemini 2.5 Pro**	**o3**	**Q**
Christopher’s Jewel	X	Exponential	X	
Pancake Sorting	X	Exponential	X	
Transposons	X	X	X	
Reversals	X	X	X	
Glushkov Problem	X	Factorial	X	
RapapportM2	X	X	X	
Rubik’s Cube 4x4x4	X	Exponential	X	
Professor Tetraminx	Exponential	Exponential	X	
Megaminx	X	X	X	
SuperCube (IHES)	X	Exponential	X	
LRX	**Poly**	Exponential	X	

Analysis

Most of the solutions that were generated by LLMs were optimal in terms of lengths, but extremely slow because of the high complexity (such solutions are highlighted with "Exponential" or "Factorial"). Models from OpenAI family have shown poor results, because of attempts to generate polynomial algorithms in all cases and strong hallucinations. Meanwhile, Gemini 2.5 Pro has generated as much solutions as Claude Opus 4.1, but struggled to find correct polynomial algorithm in case of LRX. Another useful insight that we can highlight is attempt of the Claude Opus 4.1 to generate heuristic for every solution, which is valuable, but not always correct. With rapid development of LLMs we predict that firstly we will be able to get "Exponential" algorithms in all cases and then, achieve polynomial, but sub-optimal algorithms.

Discussion

Ethics statement

LLMs were used for generating potential solutions to benchmark current state-of-art systems against human results, as well as for code assistance in human submissions.

Reproducibility statement

The present article was written using https://overleaf.comOverleaf.

The code was executed using Kaggle, Colab or GCP environments, using the following packages:

- pandas
- matplotlib
- numpy
- cayleypy
- torch (torch_xla for TPU)
- numba
- scipy

The code is available in an anonymous https://anonymous.4open.science/r/cayleyrl-neurips-2025/GitHub repository.

References

[Acevedo, Roland, and Cerf(2006)] Acevedo, O. L.; Roland, J.; and Cerf, N. J. 2006. Exploring scalar quantum walks on Cayley graphs. arXiv:quant-ph/0609234.

[Agostinelli(2019)] Agostinelli, F. e. a. 2019. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8): 356–363.

[Agostinelli(2024)] Agostinelli, F. e. a. 2024. Q* search: Heuristic search with deep q-networks. In *ICAPS Workshop on Bridging the Gap between AI Planning and Reinforcement Learning*.

[Akers and Krishnamurthy(1989)] Akers, S. B.; and Krishnamurthy, B. 1989. A group-theoretic model for symmetric interconnection networks. *IEEE transactions on Computers*, 38(4): 555–566.

[Amrutha and Srinath(2022)] Amrutha, B. V.; and Srinath, R. 2022. Deep Learning Models for Rubik's Cube with Entropy Modelling. In *ICDSMLA 2020: Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications*, 35–43. Springer Singapore.

[Bafna and Pevzner(1998)] Bafna, V.; and Pevzner, P. A. 1998. Sorting by reversals. *SIAM Journal on Discrete Mathematics*, 11(2): 224–240.

[Bamberg(2014)] Bamberg, J. e. a. 2014. Bounds on the diameter of Cayley graphs of the symmetric group. *Journal of Algebraic Combinatorics*, 40(1): 1–22.

[Bedaywi, Longtai, and Shaul(2023)] Bedaywi, M.; Longtai, T. D.; and Shaul, F. 2023. Solving the Rubik's Cube via Sequence Modeling using Transformer-based Models. `https://github.com/tedtedtedtedtedted/Solve-Rubiks-Cube-Via-Transformer/blob/main/Report.pdf`.

[Browning and Lutz(2021)] Browning, T.; and Lutz, P. 2021. Formalizing Galois Theory. *arXiv preprint arXiv:2107.10988*.

[Brunetto and Trunda(2017)] Brunetto, R.; and Trunda, O. 2017. Deep Heuristic-learning in the Rubik's Cube Domain: An Experimental Evaluation. In *ITAT*, 57–64.

[Bulteau, Fertin, and Rusu(2015)] Bulteau, L.; Fertin, G.; and Rusu, I. 2015. Pancake flipping is hard. *Journal of Computer and System Sciences*, 81(8): 1556–1574.

[Bulteau and Weller(2019)] Bulteau, L.; and Weller, M. 2019. Parameterized algorithms in bioinformatics: an overview. *Algorithms*, 12(12): 256.

[Carneiro(2018)] Carneiro, M. 2018. A Lean Formalization of Matiyasevič's Theorem. *arXiv preprint arXiv:1802.01795*.

[Chasmai(2021)] Chasmai, M. E. 2021. CubeTR: Learning to Solve the Rubik's Cube using Transformers.

[Cooperman, Finkelstein, and Sarawagi(1991)] Cooperman, G.; Finkelstein, L.; and Sarawagi, N. 1991. Applications of cayley graphs. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 8th International Conference, AAECC-8 Tokyo, Japan, August 20–24, 1990 Proceedings 8*, 367–378. Springer Berlin Heidelberg.

[del Barco et al.(2025)del Barco, Infanti, Rivas, and Schwahn] del Barco, V.; Infanti, G.; Rivas, E.; and Schwahn, P. 2025. Formalizing a classification theorem for low-dimensional solvable Lie algebras in Lean. arXiv:2505.19975.

[Demaine, Eisenstat, and Rudoy(2017)] Demaine, E. D.; Eisenstat, S.; and Rudoy, M. 2017. Solving the Rubik's Cube Optimally is NP-complete. arXiv:1706.06708.

[Diaconis(2013)] Diaconis, P. 2013. Some things we've learned (about Markov chain Monte Carlo).

[Dinur et al.(2023)Dinur, Hsieh, Lin, and Vidick] Dinur, I.; Hsieh, M. H.; Lin, T. C.; and Vidick, T. 2023. Good quantum LDPC codes with linear time decoders. In *Proceedings of the 55th annual ACM symposium on theory of computing*, 905–918.

[Egri-Nagy and Gebhardt(2016)] Egri-Nagy, A.; and Gebhardt, V. 2016. Computational Enumeration of Independent Generating Sets of Finite Symmetric Groups. arXiv:1602.03957.

[Even and Goldreich(1981)] Even, S.; and Goldreich, O. 1981. The minimum-length generator sequence problem is NP-hard. *Journal of Algorithms*, 2(3): 311–313.

[Fiat(1989)] Fiat, A. e. a. 1989. Planning and learning in permutation groups. In *30th Annual Symposium on Foundations of Computer Science*, 274–279. IEEE Computer Society.

[Gates and Papadimitriou(1979)] Gates, W. H.; and Papadimitriou, C. H. 1979. Bounds for sorting by prefix reversal. *Discrete Mathematics*, 27(1): 47–57.

[Gromada(2022)] Gromada, D. 2022. Some examples of quantum graphs. *Letters in Mathematical Physics*, 112(6): 122.

[Gromov(1993)] Gromov, M. 1993. *Geometric Group Theory: Asymptotic invariants of infinite groups*, volume 2. Cambridge University Press.

[Hannenhalli and Pevzner(1995)] Hannenhalli, S.; and Pevzner, P. A. 1995. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of IEEE 36th annual foundations of computer science*, 581–592. IEEE.

[Hannenhalli and Pevzner(1999)] Hannenhalli, S.; and Pevzner, P. A. 1999. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1): 1–27.

[He et al.(2025)He, Liang, Xu, and et al] He, Z.; Liang, T.; Xu, J.; and et al, Q. L. 2025. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. arXiv:2504.11456.

[Helfgott(2019)] Helfgott, H. A. 2019. Growth in linear algebraic groups and permutation groups: towards a unified perspective. In *Groups St Andrews 2017 in Birmingham*, volume 455, 300.

[Helfgott and Seress(2014)] Helfgott, H. A.; and Seress, Á. 2014. On the diameter of permutation groups. *Annals of mathematics*, 611–658.

[Helfgott, Seress, and Zuk(2015)] Helfgott, H. A.; Seress, Á.; and Zuk, A. 2015. Random generators of the symmetric group: diameter, mixing time and spectral gap. *Journal of Algebra*, 421: 349–368.

[Heydemann(1997)] Heydemann, M. C. 1997. Cayley graphs and interconnection networks. In *Graph symmetry: algebraic methods and applications*, 167–224. Springer Netherlands.

[Hoory, Linial, and Wigderson(2006)] Hoory, S.; Linial, N.; and Wigderson, A. 2006. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4): 439–561.

[Jerrum(1985)] Jerrum, M. R. 1985. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36: 265–289.

[Johnson(2021)] Johnson, C. G. 2021. Solving the Rubik's cube with stepwise deep learning. *Expert Systems*, 38(3): e12665.

[Khandelwal, Sheth, and Agostinelli(2024)] Khandelwal, V.; Sheth, A.; and Agostinelli, F. 2024. Towards Learning Foundation Models for Heuristic Functions to Solve Pathfinding Problems. arXiv:2406.02598.

[Knuth(1991)] Knuth, D. E. 1991. Efficient representation of perm groups. *Combinatorica*, 11(1): 33–43.

[Korf(1997)] Korf, R. E. 1997. Finding optimal solutions to Rubik's Cube using pattern databases. In *AAAI/IAAI*, 700–705.

[Larsen(2003)] Larsen, M. 2003. Navigating the Cayley graph of $\mathrm{SL}_2(\mathbb{Z}/p\mathbb{Z})$. *Int. Math. Res. Not.*, (27): 1465–1471.

[Loeffler and Stoll(2025)] Loeffler, D.; and Stoll, M. 2025. Formalizing zeta and L-functions in Lean. *Annals of Formalized Mathematics*, Volume 1.

[Manthe(2025)] Manthe, S. 2025. A formalization of Borel determinacy in Lean. arXiv:2502.03432.

[McAleer(2019)] McAleer, S. e. a. 2019. Solving the rubik's cube with approximate policy iteration. In *International Conference on Learning Representations*.

[Noever and Burdick(2021)] Noever, D.; and Burdick, R. 2021. Puzzle solving without search or human knowledge: An unnatural language approach. arXiv:2109.02797.

[Otte(2025)] Otte, P. 2025. Tutte's theorem as an educational formalization project. arXiv:2504.18146.

[Pan and Kondor(2021)] Pan, H.; and Kondor, R. 2021. Fourier Bases for Solving Permutation Puzzles. In *International Conference on Artificial Intelligence and Statistics*, PMLR, 172–180.

[Paster et al.(2023)Paster, Santos, Azerbayev, and Ba] Paster, K.; Santos, M. D.; Azerbayev, Z.; and Ba, J. 2023. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. arXiv:2310.06786.

[Petit and Quisquater(2011)] Petit, C.; and Quisquater, J. J. 2011. Rubik's for cryptographers. Cryptology ePrint Archive.

[Rokicki(2014)] Rokicki, T. e. a. 2014. The diameter of the rubik's cube group is twenty. *SIAM REVIEW*, 56(4): 645–670.

[Ruiz(2024)] Ruiz, F. J. e. a. 2024. Quantum Circuit Optimization with AlphaTensor. `https://arxiv.org/abs/2402.14396`. arXiv:2402.14396.

[Sarkar and Adhikari(2024)] Sarkar, R. S.; and Adhikari, B. 2024. Quantum circuit model for discrete-time three-state quantum walks on Cayley graphs. *Physical Review A*, 110(1): 012617.

[Saxton et al.(2019)Saxton, Grefenstette, Hill, and Kohli] Saxton, D.; Grefenstette, E.; Hill, F.; and Kohli, P. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. arXiv:1904.01557.

[Sims(1970)] Sims, C. C. 1970. Computational methods in the study of permutation groups. In *Computational problems in abstract algebra*, 169–183. Pergamon.

[Swan(2017)] Swan, J. 2017. Harmonic analysis and resynthesis of Sliding-Tile Puzzle heuristics. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, 516–524. IEEE.

[Takano(2021)] Takano, K. 2021. Self-Supervision is All You Need for Solving Rubik's Cube. arXiv:2106.03157.

[Tao(2015)] Tao, T. 2015. *Expansion in finite simple groups of Lie type*, volume 164. American Mathematical Soc.

[The mathlib Community(2020)] The mathlib Community. 2020. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '20)*, 123–135.

[Wieser and Song(2022)] Wieser, E.; and Song, U. 2022. Formalizing Geometric Algebra in Lean. *Advances in Applied Clifford Algebras*, 32(3).

[Zémor(1994)] Zémor, G. 1994. Hash functions and Cayley graphs. *Designs, Codes and Cryptography*, 4(3): 381–394.

[Zheng, Han, and Polu(2021)] Zheng, K.; Han, J. M.; and Polu, S. 2021. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. arXiv:2109.00110.

Appendix

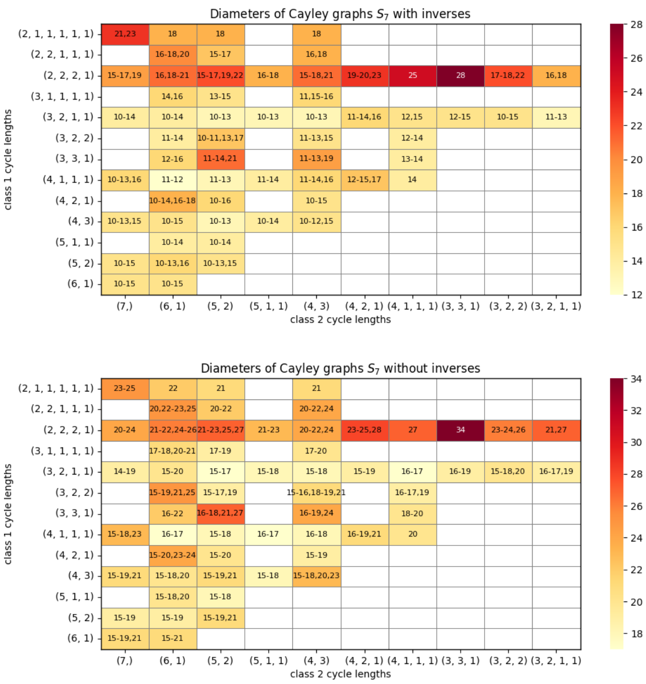

Figure 1: Diameters of all possible Cayley graphs for S_7 generated by two permutations with/without their inverses.

LLM prompt for LRX algorithm generation

You are given sequences consisting of only 0 and 1. The task is to sort given input sequences using only three allowed transformations L,R,X: L - left cyclic shift, R - right cyclic shift, X - swap positions 0 and 1: X(A,B,C,D,. . . .) = (B,A,C,D,. . . .), Left shift: L(A,B,C,D,. . . U,V,Z) = (B,C,D,. . . U,V,Z,A), and Right shift: R(A,B,C,D,. . . U,V,Z) = (Z,A,B,C,D. . . ,U,V). It is known that any sequence can be transformed to the sorted one using L,R,X transformations.

Goal is to find the shortest sorting sequence of L,R,X permutations - for each entry in test data. Your score - is the sum of lengths

Generate polynomial algorithm on Python that will solve the puzzle with maximum optimality that you can achieve

Distribution of diameter over conjugacy classes pairs - involutions strikes

Figure 1 represents diameters dependence on the conjugacy classes of generators obtained by exhaustive search: one generator from one class, another from another. Presented figure is for S_7. The values in each cell represent all values of the diameters found for corresponding pair of classes and the heatmap coloring is done with respect to the maximal found diameter for corresponding pair of classes. Color is white if S_n is not generated by any pair of elements from the above classes. Due to symmetry we are showing only relevant part of data, not duplicating $g1, g2$ with $g2, g1$.

One can see that large diameters appears when one of the classes is involution, and another has rather short cycles in decomposition. Which is simple to expect, because the longer cycles are present - means individual generators would have higher degree and allow to create more words like: $XX...XX$. More words of fixed length k one has - the less diameter can be, since words exhausts finite space of group elements earlier. So naively it is easy to expect that generators related to as small degree as possible are good candidates. Exhaustive search confirms it for many cases. So:

The generating set with largest diameter for S_n contains an involution (at least for infinite number of n). Both for directed and undirected cases.

We performed similar exhaustive search up to $n \leq 9$ and randomized search up to 12 – for the pairs of generators and for both undirected and directed cases, results are consistent with the conjecture. Similar heatmap plots for other n can be found in full version the manuscript; or up to $n \leq 7$ in the https://www.kaggle.com/code/fedmug/diameter-visualizationsnotebook.

Table 2: Summary of properties of Cayley graphs

Gene-	Gro-	Dia-	Growth							Anti-	Algo-	Met-	Spec-	Mixing
rators	**up**	**meter**	**PDF**	**F-la**	**Mean**	**Mode**	**Var**	**Skew**	**Kurt**	**podes**	**rithm**	**ric**	**trum**	**Time**
Coxeter	S_n	$\frac{n(n-1)}{2}$	Gauss	+	n(n-1)/4	n(n-1)/4	$+$	$\to 0$	$\to 0$	1	Bubble	$+$	?	?
Cyclic Coxeter	S_n	$\lfloor \frac{n^2}{4} \rfloor$	Gauss◆	?	$0.17(n^2-n+1)$◆	≈Mean◆	?	$\to 0$◆	$\to 0$◆	1|2◆	?	+	Wig◆	?
LRX	S_n	$\frac{n(n-1)}{2} *$	Gumbel◆	?	$\approx 0.38n^2-n$◆	$\approx 0.39n^2-n$◆	?	$\to -0.7$◆	$\to 3.3$◆	◆	◆	?	Uni◆	$> n^3$◆
LX-Glushkov	S_n	$\frac{3n^2-8n+9\|12}{4}$◆	Gumbel◆	Fib /?	$\approx 0.57n^2-2n$◆	$\approx 0.57n^2-1.6n$◆	?	$\to -0.7$◆	$\to 0.5$◆	$*$	◆	?	?	?
LARX	S_n	$\frac{n^2-2\|5}{2}$◆	?	?	$0.4n^2-0.7n$ ◆	≈Mean◆	?	?	?	+◆|?	?	?	?	?
LARX+I	S_n	$\frac{n(n+6)-12\|19}{4}$◆	?	?	$\approx \frac{n(n+1)}{4}$◆	≈Mean◆	?	?	?	$+$◆	?	?	?	?
LSL	S_n	$\frac{n(n-3)}{2}+3$◆	?	?	$\approx 0.4n^2-1.5n$◆	≈Mean◆	?	?	?	$+$◆	?	?	?	?
LSL+I	S_n	$\frac{n(n+4)}{4}-3\|4.25$◆	?	?	$\approx 0.2n^2$	$\approx 0.2n^2$	?	?	?	?	?	?	?	?
3-cyc	A_n	$\lfloor \frac{n}{2} \rfloor$	?	$+-$	$\approx D-0.5\|1$◆	$D-0\|1$◆	?	?	?	$+$	$+$	$+$	Int	?
(0ij)	A_n	$\lfloor \frac{3(n-1)}{4} \rfloor$	?	?	$\approx 0.55n$◆	$\approx Mean$◆	?	?	?	?	?	?	?	?
(01i)	A_n	$\frac{3n-5}{2}+\frac{i^n+(-i)^n}{4}$◆	?	?	$\approx n-2$◆	$n-1$◆	?	?	?	?	$+$◆	?	?	?
(01i)I	A_n	$\lfloor \frac{3n-6}{2} \rfloor$ ◆	?	?	$\approx n+1.25ln(n)+\ldots$◆	≈Mean◆	◆	◆	◆	◆	◆^{O}	◆	Int◆	?
(i,i+1,i+2)	A_n	$\lceil \frac{n^2+1}{4} \rceil$◆	Gauss◆	?	$\approx D/2$◆	$\approx D/2$◆	$\approx \frac{n^3}{100}$	$\to 0$◆	$\to 0$◆	?	?	?	?	?
(i,i+1,i+2)I	A_n	$\lfloor \frac{n(n-1)}{4} \rfloor$◆	Gauss◆	?	$\approx D/2$◆	$\approx D/2$◆	$\approx \frac{n^3}{100}$◆	$\to 0$◆	$\to 0$◆	$+$◆	?	?	?	?
(i...i+3)	S_n	$\approx 0.3n^2$◆	?	?	$\approx 0.16n^2$◆	$\approx 0.15n^2$◆	?	≈ 0◆	≈ 0◆	?	?	?	?	?
(i...i+3)I	S_n	$\approx 0.16n^2$◆	?	?	$\approx 0.036n^2$◆	$\approx 0.045n^2$◆	?	?	?	?	?	?	?	?
(i,i+1,i+2)C	A_n	$\approx \frac{n(n+2)}{8}+I$◆	?	?	$\approx 0.085n^2$◆	$\approx 0.065n^2$◆	?	?	?		?	?	?	?
(i,i+1,i+2)CI	A_n	$\lfloor \frac{n^2}{8} \rfloor$ ◆	?	?	$\approx 0.08n^2$◆	$\approx 0.086n^2$◆	?	?	?	$+$◆	?	?	?	?
(i...i+3)C	S_n	?	?	?	?	?	?	?	?	?	?	?	?	?
(i...i+3)CI	S_n	?	?	?	?	?	?	?	?	?	?	?	?	?
Pref.cyc	S_n	$n-1$												
Pref.cyc+I	S_n	$n-1$◆	?	?							?	?	?	?
Down.cyc	S_n	$n-1$◆	?	Stirling	$\approx 0.86n$◆	≈Mean◆	?	?	?	$+$◆	?	?	?	?
Down.cyc+I	S_n	$n-1$◆	?	?	$\approx 0.75n$◆	≈Mean◆	?	?	?	$+$◆	?	?	?	?
Inc.3cyc	A_n	$n-1\|2$◆	?	?	$\approx 0.5n$◆	≈Mean◆	?	?	?	$+-$◆	?	?	?	?
Inc.4cyc	S_n	$n-1$◆	?	?	$\approx 0.5n$◆	≈Mean◆	?	?	?	?	?	?	?	?
RapaportM1	S_n	$\lceil \frac{3n}{2} \rceil$◆	?	?	$\approx 1.4n$◆	≈Mean◆	?	?	?	?	?	?	?	?
RapaportM2	S_n	$\approx \frac{n^2+n}{2}$ ◆	?	?	$\approx \frac{n^2-3n}{2}$◆	≈Mean◆	?	$\to -0.6$◆	$\to 0.5$◆	?	?	?	?	?
Globes n/1	+	$\lceil \frac{5n+12}{4} \rceil$◆	?	?	$\approx n+1$◆	$\approx n+1$◆	?	?	?	?	?	?	?	?
3PancakeS_1	S_n	$\frac{3n(n+2)-48+I}{8}$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_2	S_n	◆	?	?	◆		?	?	?		?	?	?	?

Continued on next page

Gene-rators	Gro-up	Dia-meter	Growth							Anti-podes	Algo-rithm	Met-ric	Spec-trum	Mixing Time
			PDF	F-la	Mean	Mode	Var	Skew	Kurt					
3PancakeS_3	S_n	◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_4	S_n	◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_5	S_n	◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_6	S_n	$\frac{3n(n+2)-48}{8}$ ◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_7	S_n	◆	?	?	◆		?	?	?		?	?	?	?
Pancake	S_n	$\approx 1.2n?$	?	?	$< \frac{17n}{12}$	$\approx n/2*$	$\approx 0.2n?$	?	?	$+-$	$NP/2?$	$-$	?	?
Reversals	S_n	$n-1$	?	?	$\approx n/2$	$\approx Mean?$	$\approx 0.05n$?	?	?	$+$	$NP/1.5$	$-$	?	?
sReversals	B_n	$n-1$	?	?	$\approx n - \frac{nlogn}{2}$	$\approx Mean?$	$\approx 0.05n$	?	?	$+-$	$+$	$+-$	?	?
Transposons	S_n	$\lceil \frac{n+1}{2} \rceil *$	?	?	$\Theta(n)$	$\Theta(n)$		?	?	$-$	$NP/1.375$	$-$	?	?
(i,j)	S_n	$n-1$	?	Stirling	$\approx n - lnn$	$\approx n - lnn$	$\approx lnn$	$\approx \frac{1}{\sqrt{lnn}}$	$\approx \frac{1}{lnn}$	$+$	$+$	$+$	Int	nlnn
$(1,i)$(Star)	S_n	$\left\lfloor \frac{3(n-1)}{2} \right\rfloor$	?	?	$\approx n - lnn$	$\approx n - lnn$	$\approx lnn$			$+$	$+$	$+$	Int	?
G-star transp.	S_n	$+$	?	?	?	?	?	?	?	?	?	?	?	?

End of table

For the table for the coset graphs we use same notations modulo two exceptions

1. "Coset" - type of the coset, currently we mainly work with "Bin", which is $S_n/(S_{n//2} \times S_{n-n//2})$ ("Grassmanian over F_1"), or, in the simple language, - nodes are vectors with only 0,1 coordinates and with $n//2$ zeros
2. "God's number" - instead of diameter we consider the largest distance to some selected node in the graph. We take vectors with zeros first, then units (i.e. sorted vectors) as our default choice for "Bin" for such initial state.

Table 3: Summary of properties of Schreier graphs

Gene-rators	Co-set	God's number	Growth							Anti-podes	Algo-rithm	Met-ric	Spec-trum	Mixing Time
			PDF	F-la	Mean	Mode	Var	Skew	Kurt					
LRX	Bin	$\frac{n(3n-4)+32-2(n\%4)}{16}$ ◆	?	?	$\approx 0.16n^2$	$\approx 0.16n^2$	?	$\approx -.5$◆	$\to 3$◆	$+-$◆	$+$◆	?	?	?
Pancake	Bin	$n-1\|2$◆	?	$+$◆	$\approx n/2$◆	$n//2 - I(n\%4)$◆	?	$\to 0$◆	$\to 0$◆	$+$◆	$+$◆	?	?	?
Transposons	Bin	$\lfloor \frac{n}{2} \rfloor$◆	Gauss◆	$+$◆	$n/4$◆	$(n+1)//4$◆	?	$\to 0$◆	$\to 0$◆	$+$◆	?	?	?	?
Reversals	Bin	Growth coincides with transposons												
RapaportM1	Bin	$n-1$◆	?	?	$\approx \frac{3n-4}{4}$◆	$\approx Mean$◆	?	?	?	$+-$◆	?	?	?	?
RapaportM2	Bin	$\frac{n(n+1)}{4} - 3\|4.75$◆	?	?	$\approx 0.21n$◆	≈Mean◆	?	≈ 0.6◆	≈ 0◆	$+$◆	?	?	?	?
(i,i+1,i+2)	Bin	$1/8n^2 + I(\%4)$◆	?	?	$\approx 0.06n^2$◆	$\approx 0.06n^2$◆	?	$\to 0$◆	$\to 0$◆	$+-$◆	?	?	?	?
(i...i+3)	Bin	$n^2/2 + I(\%6)$◆	?	?	$\approx 0.04n^2$◆	$\approx 0.04n^2$◆	?	$\to 0$◆	$\to 0$◆	$+-$◆	?	?	?	?
(i...i+4)	Bin	$9/16n^2 + I(\%8)$◆	?	?	$\approx 0.03n^2$◆	$\approx 0.03n^2$◆	?	$\to 0$◆	$\to 0$◆	$+-$ ◆	?	?	?	?
3PancakeS_1	Bin	$\frac{3n^2+36}{16} + I$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_2	Bin	$\approx \frac{n(n-2)}{4}$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_3	Bin	$\frac{n(n+14)-I}{8}$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_4	Bin	$\approx \frac{5n^2}{32}$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_5	Bin	$\approx \frac{47n^2}{128}$◆	?	?	◆		?	?	?		?	?	?	?

Continued on next page

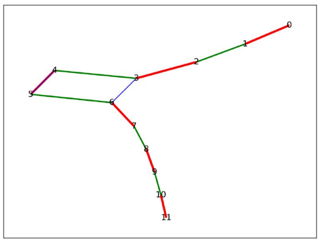

Figure 2: Three generators of S_{12} represented graphically. Pattern - "square with whiskers". Three involutions - edges orientations are not necessary.

To the best of our knowledge these are the largest diameters known so far. We consider both standard undirected and directed cases (meaning that generators are not necessarily inverse closed). For the latter case the same diameters up to $n \leq 7$ were found in [Egri-Nagy and Gebhardt(2016)] (table 4), but our result extends to much larger n.

The diameters provided below are largest possible or at least within say 5% from them. It is impossible to make exhaustive search even for such values of n, so we cannot exclude the chance that large diameters exists, though it seems unlikely to us that they will be significantly larger than presented here. Anyway we hope our results may stimulate that research.

Gene-rators	**Co-set**	**God's number**	**Growth**							**Anti-podes**	**Algo-rithm**	**Met-ric**	**Spec-trum**	**Mixing Time**
			PDF	**F-la**	**Mean**	**Mode**	**Var**	**Skew**	**Kurt**					
3PancakeS_6	Bin	$\frac{3n^2}{16}+I$◆	?	?	◆		?	?	?		?	?	?	?
3PancakeS_7	Bin	$\frac{5n^2+58n-64+I}{72}$	◆ ?	?	◆		?	?	?		?	?	?	?
End of table														

Graphical visual representation for any generators - pattern "square with whiskers"

Here we describe a very simple graphical visual representation of elements (e.g. generators) of permutation groups, and present a pattern "square with whiskers" which corresponds to most of the largest diameters found for $n \leq 15$.

Step 1. Single permutation. Each permutation defines a directed graph on n nodes in a natural and obvious way – if $p(i) = j$ let us connect $i \to j$. (That can be said in the other words - take a permutation matrix and consider it as adjacency matrix of a directed graph). Clearly if permutation is involution - then orientation of edges is unnecessary - if $i \to j$, then $j \to i$ (in matrix language - permutation matrix is symmetric).

Step 2. Many permutations - use colors. Consider several permutations and just use the same construction but use different colors to represent edges coming from different permutations.

Thus for any sequence of permutations (generators) we constructed a directed multi-colored graph on n nodes.

The code for the visualization can be found e.g. in the https://www.kaggle.com/code/olegpushs/draw-edges notebook.

The figure 2 presents an example of such visualization and also presents an example of the pattern which we call "square with whiskers" - there is one 4-cycle (square) and two branches going out of his corners.

Let us call the generators to follow "square with whiskers" pattern if underlying undirected graph (forgetting colors and multi-edges) is of that type - one 4-cycle (square) and two branches.

The generators with maximal (or nearly) diameter for S_n/A_n follow "square with whiskers" pattern (at least for infinite number of n). We expect that to be true for the both for undirected and directed case of Cayley graphs.

The computations up to $n \leq 15$ described above supports that conjecture.

Largest diameters found (and known) for $n <= 15$n¡=15

We conducted extensive search for generators producing large diameters for small n, remarkable patterns showed up - that will be discussed in the next section, here we just present the diameters and the corresponding generators, and organization of the experiments. But already here it is worth to highlight that all found generators are related to involutions.

Maximal diameter for Cayley graph of the group S_n (undirected graph)		
n	Maximal diameter	Example of a set of generators
3	3	$[0,2,1]$, $[2,1,0]$
4	6	$[2,1,0,3]$, $[3,0,2,1]$, $[1,3,2,0]$
5	10	$[4,0,1,2,3]$, $[1,2,3,4,0]$, $[0,1,3,2,4]$
6	15 (16)	$[4,5,3,2,0,1]$, $[2,5,0,3,1,4]$, $[2,4,0,3,5,1]$ ($[[0,1,2,3,5,4]$, $[0,2,1,4,3,5]$, $[1,0,3,2,5,4]]$)
7	28 (30)	$[1,0,3,2,5,4,6]$, $[2,6,5,3,1,0,4]$, $[5,4,0,3,6,2,1]$ ($[0,1,3,2,4,6,5]$, $[0,4,6,5,1,3,2]$, $[6,1,3,2,5,4,0]$)
8	33 (39)	$[3,7,5,6,0,2,4,1]$, $[4,7,5,0,6,2,3,1]$, $[1,0,3,2,5,4,6,7]$ ($[0,1,2,3,5,4,7,6]$, $[0,1,3,2,6,7,4,5]$, $[7,3,6,1,4,5,2,0]$)
9	(52)	($[8,5,2,7,4,1,6,3,0]$, $[1,2,0,5,3,4,8,6,7]$, $[2,0,1,4,5,3,7,8,6]$)
10	(77)	($[0,1,2,3,5,4,7,6,9,8]$, $[1,0,3,2,5,4,8,9,6,7]$, $[0,6,4,8,2,5,1,7,3,9]$)
11	(85)	($[1,0,3,2,5,4,7,6,9,8,10]$, $[0,3,4,1,2,6,5,8,7,10,9]$, $[0,4,3,2,1,5,6,7,8,9,10]$)
12	(95)	($[1,0,3,2,5,4,7,6,9,8,11,10]$, $[0,2,1,5,6,3,4,8,7,10,9,11]$, $[0,1,2,6,5,4,3,7,8,9,10,11]$)
13	(111)	($[1,0,2,5,4,3,7,6,9,8,11,10,12]$, $[0,2,3,4,1,6,5,8,7,10,9,12,11]$, $[0,4,1,2,3,6,5,8,7,10,9,12,11]$)
14	(132)	($[1,0,3,2,5,4,7,6,9,8,11,10,13,12]$, $[0,2,1,4,3,6,5,8,7,10,9,12,11,13]$, $[0,1,2,3,5,4,6,7,8,9,10,11,12,13]$)
15	(148)	($[1,0,3,2,5,4,7,6,9,8,11,10,13,12,14]$, $[0,2,1,4,3,6,5,8,7,10,9,12,11,14,13]$, $[0,1,2,3,4,8,7,6,5,9,10,11,12,13,14]$)

Maximal diameter for Cayley graph of the group S_n (oriented graph)		
n	Maximal diameter	Example of a set of generators
3	3	$(01), (02)$
4	7	$(01), (123)$
5	14	$(01)(23), (0314)$
6	18	$(01)(23)(45), (012)(34)$
7	34	$(01)(23)(45), (052)(146)$
8	44	$(01)(23)(45), (1736)(25)$
9	61	$(01)(23)(45), (3647)(12)(58)$
10	83	$(01)(23)(45)(67)(89), (185)(237)(469)$
11	93	$(01)(23)(45)(67)(89), (1528)(47)(6,10)$
12	106	$(01)(23)(45)(67)(89), (1,11,9,10)(04)(28)(36)$
13	147	$(01)(23)(45)(67)(89), (1,11,2,12)(34)(56)(78)(9,10)$

For the directed case (i.e. not inverse closed generators) the search has been organized as follows - for small $n = 3, 4$ we considered several possible numbers of generators - from 2 to 5, it was observed that largest diameter is observed for 2 generators - which is not surprising, since less generators - less words can one generate and large diameter can potentially be. We continue search with 2 generators for n up to 13. The search has been - exhaustive up to $n \leq 9$, and randomized search after. We did not search all possible pairs, but relied on a simple fact that conjugacy of all generators produces an isomorphic graph, so we say first generator can be taken as a unique representative of conjugacy class, with the loop over conjugacy classes - that of course significantly reduce the search space. For the randomized search we sampled the second generator from each conjugacy class uniformly. To reduce search further for $n \geq 12$ we mostly considered conjugacy classes for the first generator to be involutions, and used guesses from previously observed patterns.

Acknowledgments

The study was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002)