

# Exact Counting and Uniform Sampling of Pressing Sequences on Paths and Cycles

Alper Ferudun\*

25 October 2026

## Abstract

A pressing sequence of a black-and-white graph repeatedly presses a black vertex, complements its closed neighborhood, and ends at the all-white edgeless graph. We give a polynomial-time algorithm that exactly counts these sequences when the underlying graph has maximum degree two. The algorithm also counts full-length sequences, detects when none exist, and supports exact uniform sampling. Its main ingredient is a last-vertex recurrence for intervals of a path, gated by the determinant of a binary tridiagonal matrix. For a cycle, nullity at most two reduces the remaining cases to one or two unpressed vertices. Preprocessing uses  $O(n^3)$  arithmetic operations on integers with  $O(n \log(n+1))$  bits, and a sample can be obtained by unranking in  $O(n^2)$  arithmetic operations after one uniform integer draw. In particular, this gives a polynomial-time answer to the full-length path-counting question stated by Miklós in 2023. The general-graph complexity questions and rapid mixing of the previously studied Markov chain are not resolved.

## 1 Definitions and main result

Let  $G$  be a finite labeled simple graph, with each vertex colored black or white. Write  $b_v = 1$  for black and  $b_v = 0$  for white. Pressing a black vertex  $v$  toggles the colors of its neighbors, complements adjacency among those neighbors, and leaves  $v$  isolated and white. Equivalently, one complements the whole closed neighborhood, regarding colors as loops. A pressed vertex can be deleted immediately: it cannot subsequently be pressed or affect another press.

A sequence of distinct vertices is *legal* if every vertex is black when pressed. It is *successful* if the residual graph is all-white and edgeless. The empty sequence is successful for an initially all-white edgeless graph. A *full* sequence is legal and presses every vertex; it is automatically successful. We count labeled sequences, not equivalence classes under graph automorphisms. Loops encoding colors are excluded when speaking of degree.

Cooper and Davis developed the binary-matrix description of this game and asked about counting and approximation [1]. Cooper’s problem list [7], recorded as AMR-030-0046 in UnsolvedMath 1.6.0 [8], asks these questions for general bicolored graphs. Bixby, Flint, and Miklós proved connectivity of a local-edit graph of successful sequences on paths [2]. That connectivity result does not give a bound on mixing time.

In a 2023 Dagstuhl report, Miklós specifically asked whether the number of full successful sequences of a black-and-white path can be computed in polynomial time [3, Section 4.8, p. 71]. The report defines those sequences as permutations of all vertices. We keep this full-length convention separate from successful sequences that leave isolated white vertices unpressed.

**Theorem 1.** *Let  $G$  be an  $n$ -vertex bicolored graph whose underlying simple graph has maximum degree at most two. Both the number of successful pressing sequences and the number of*

---

\*Mercury Software GmbH. Email: [alper@mercurycodelab.com](mailto:alper@mercurycodelab.com). GitHub: [AlperTheKing](https://github.com/AlperTheKing).

full sequences can be computed exactly using  $O(n^3)$  integer arithmetic operations and  $O(n^3)$  binary field operations. Every integer used has  $O(1 + n \log(n + 1))$  bits. The tables require  $O(n^2)$  such integer entries.

For either convention, zero count is detected. When the count is positive, a uniformly random sequence can be generated exactly, using a uniform integer below that count followed by  $O(n^2)$  integer arithmetic operations. Counting takes deterministic polynomial bit time, and exact sampling takes expected polynomial bit time in the fair-random-bit model.

The proof includes arbitrary colorings and singular augmented adjacency matrices. Uniquely pressable graphs and autonomous pressing posets have been studied separately [4, 5]; we do not assume either property. The theorem settles the counting alternative in the path question of [3], and supplies an exact sampler on the same class. It neither classifies the counting problem on arbitrary graphs nor proves rapid mixing of the chain in [2]. Recent hardness results for other genome rearrangement models [6] concern different operations and do not by themselves settle this pressing problem.

## 2 A restriction and rank lemma

All matrix operations in this section are over  $\mathbb{F}_2$ . Let  $A = A(G)$  be the augmented adjacency matrix:  $A_{vv} = b_v$ , and  $A_{uv} = 1$  for an edge  $uv$ ,  $u \neq v$ . If  $v$  is black, place it first and write

$$A = \begin{pmatrix} 1 & a^\top \\ a & B \end{pmatrix}.$$

After pressing and deleting  $v$ , the remaining matrix is  $B + aa^\top$ . Elementary congruence gives

$$A \sim \begin{pmatrix} 1 & 0 \\ 0 & B + aa^\top \end{pmatrix}. \quad (1)$$

In particular, each legal press reduces rank by one. The resulting rank characterization is standard [1, Corollary 4]; we include the needed restriction statement explicitly.

**Lemma 2.** *Let  $S \subseteq V(G)$ , and let  $\sigma$  be an ordering of all vertices of  $S$ . The sequence  $\sigma$  is legal in  $G$  if and only if it is a full legal sequence for the induced bicolored graph  $G[S]$ . If it is legal, then  $\det A[S] = 1$ . Moreover, a legal sequence pressing exactly  $S$  is successful in  $G$  if and only if  $|S| = \text{rank } A$ .*

*Proof.* At every step, the update to entries indexed by the surviving vertices of  $S$  is  $A_{ij} \mapsto A_{ij} + A_{iv}A_{vj}$ . It uses no entry indexed outside  $S$ . Induction therefore identifies the colors and edges on the surviving part of  $S$  with those in the induced problem. This proves the legality equivalence. Eliminating all of  $S$  inside  $A[S]$  uses unit pivots only, so its determinant is one.

Applying (1) successively to the full matrix shows that the residual matrix has rank  $\text{rank } A - |S|$ . The residual graph is all-white and edgeless exactly when this matrix is zero, equivalently when its rank is zero.  $\square$

For a bicolored graph  $H$ , let  $F(H)$  denote the number of its full legal sequences, with  $F(\emptyset) = 1$ . It follows that  $F(H) = 0$  if  $\det A(H) = 0$ . Nonsingularity alone does not guarantee  $F(H) > 0$ : an all-white two-vertex path has nonsingular matrix but no legal first move.

**Lemma 3.** *If  $H \neq \emptyset$  and  $\det A(H) = 1$ , then*

$$F(H) = \sum_{v \in V(H)} F(H - v).$$

*Proof.* Fix the last vertex  $v$ . The earlier moves form a full legal order on  $H - v$  by Lemma 2. Conversely, any such order is legal in  $H$ . Its pivots are all one, so the single residual diagonal entry is  $\det A(H) = 1$ , by repeated block determinant identities. Thus  $v$  can be pressed last. These constructions are inverse, and different last vertices give disjoint sets of orders.  $\square$

When  $H$  is a disjoint union of graphs  $H_1, \dots, H_t$ , their full legal orders can be interleaved freely:

$$F(H) = \binom{m_1 + \dots + m_t}{m_1, \dots, m_t} \prod_{\ell=1}^t F(H_\ell), \quad m_\ell = |V(H_\ell)|. \quad (2)$$

Pressing one component never changes another, so this is a bijective count, including components for which the count is zero.

### 3 The interval recurrence and paths

Consider a path with ordered vertices  $1, \dots, n$ . For a nonempty interval  $[i, j]$ , write

$$D(i, j) = \det A[i, j] \in \mathbb{F}_2, \quad F(i, j) = F(G[i, j]).$$

Set  $D(i, i-1) = F(i, i-1) = 1$ . Expanding the tridiagonal determinant gives the continuant recurrence

$$D(i, i) = b_i, \quad D(i, j) = b_j D(i, j-1) + D(i, j-2) \quad (j > i). \quad (3)$$

The addition here is modulo two; the subsequent counts are ordinary nonnegative integers.

**Proposition 4.** For  $i \leq j$ ,

$$F(i, j) = \begin{cases} 0, & D(i, j) = 0, \\ \sum_{v=i}^j \binom{j-i}{v-i} F(i, v-1) F(v+1, j), & D(i, j) = 1. \end{cases} \quad (4)$$

*Proof.* The zero case follows from Lemma 2. Otherwise apply Lemma 3. Deleting  $v$  splits the induced path into its left and right intervals. Formula (2) counts their orders and interleavings.  $\square$

Increasing interval length evaluates all entries, using  $O(n^2)$  determinant entries and  $O(n^3)$  summands. This already gives the full-length answer  $F(1, n)$ , including zero.

To count every successful sequence, let  $r = \text{rank } A$ . For  $n \geq 1$ , a kernel vector  $x$  is determined by  $x_1$ : the first row gives  $x_2 = b_1 x_1$  when  $n > 1$ , and each interior row gives  $x_{k+1} = x_{k-1} + b_k x_k$ . The last row can impose an additional condition but cannot increase the number of free coordinates. For  $n = 1$  the same nullity bound is immediate. Thus  $n - r \leq 1$ .

If  $r = n$ , every successful sequence is full. If  $r = n - 1$ , exactly one vertex  $v$  remains unpressed. Its complement consists of two intervals. Lemma 2 and (2) therefore give

$$\mathcal{N}(G) = \sum_{v=1}^n \binom{n-1}{v-1} F(1, v-1) F(v+1, n) \quad \text{when } r = n - 1. \quad (5)$$

No extra color condition on the unpressed vertex is needed: its residual matrix has rank zero by the lemma.

For example, the two-vertex path with both vertices black has two successful sequences, each of length one, and no full sequence. The one-vertex white path has one successful sequence, the empty sequence, but no full sequence. These examples explain why the two counting conventions must not be conflated.

## 4 Cycles and disjoint unions

Let a component be a simple cycle  $C_n$ ,  $n \geq 3$ , with a fixed cyclic labeling  $1, \dots, n$ . For every proper cyclic interval  $I$ , compute  $F(I)$  and  $D(I)$  by (3)–(4). There are  $n(n-1)$  nonempty proper intervals, so this still uses  $O(n^3)$  summands. In implementations they may be indexed by a starting vertex and a length less than  $n$ .

Put  $r = \text{rank } A(C_n)$ . The interior equations  $x_{k+1} = x_{k-1} + b_k x_k$ ,  $2 \leq k \leq n-1$ , determine a kernel vector from  $x_1, x_2$ . The remaining two equations impose constraints, so  $n - r \leq 2$ . The following three cases exhaust all possibilities.

If  $r = n$ , the number of successful sequences, which are full, is

$$\mathcal{N}(C_n) = F(C_n) = \sum_{v=1}^n F(C_n - v), \quad (6)$$

by Lemma 3; every summand is a proper interval entry.

If  $r = n - 1$ , the same numerical expression counts successful sequences:

$$\mathcal{N}(C_n) = \sum_{v=1}^n F(C_n - v). \quad (7)$$

Here  $v$  is the unpressed vertex, rather than the last pressed one. The full count is zero.

If  $r = n - 2$ , choose the unordered unpressed pair  $\{u, v\}$ . Its complement consists of two possibly empty cyclic intervals  $I_{uv}, J_{uv}$ , of total size  $n - 2$ . Then

$$\mathcal{N}(C_n) = \sum_{\{u,v\} \subset V(C_n)} \binom{n-2}{|I_{uv}|} F(I_{uv}) F(J_{uv}). \quad (8)$$

Every counted order is legal by restriction and independent interleaving. Since it presses  $r$  vertices, its residual two-by-two matrix is zero. Conversely every successful order determines a unique unpressed pair and a unique pair of interval orders and shuffle. The formula is symmetric in the two intervals, and each unordered pair is counted once. The full count is again zero. For the all-black triangle the augmented matrix has rank one; (8) correctly gives three one-move sequences.

Finally, every graph of maximum degree two is a disjoint union of paths, cycles, and isolated vertices. Let its components be  $G_1, \dots, G_t$ , and let  $r_\ell = \text{rank } A(G_\ell)$ . Successful orders in each component have length  $r_\ell$ . Consequently

$$\mathcal{N}(G) = \binom{r_1 + \dots + r_t}{r_1, \dots, r_t} \prod_{\ell=1}^t \mathcal{N}(G_\ell). \quad (9)$$

For full sequences use (2). The graph with no vertices has count one under both conventions.

## 5 Complexity and exact sampling

Binomial coefficients up to order  $n$  are precomputed by Pascal's recurrence. All interval tables take  $O(n^3)$  arithmetic operations in total: the sum of the cubes of the component sizes is at most  $n^3$ . A cycle needs at most  $O(n^2)$  additional terms in (8); paths need  $O(n)$  terms in (5). Gaussian elimination over  $\mathbb{F}_2$  computes all component ranks within  $O(n^3)$  field operations.

Every successful sequence has length  $r = \text{rank } A$ , so the total number is at most  $n!/(n-r)! \leq n!$ . Full counts and the nonnegative branch weights in all formulas satisfy the same bound. The binomial entries also have  $O(1 + n \log(n+1))$  bits. Thus the arithmetic operation

bounds imply polynomial bit time; for example, schoolbook integer arithmetic gives a sufficient  $O(n^5 \log^2(n+1))$  preprocessing bound for  $n \geq 2$ . The integer tables occupy  $O(n^2)$  entries.

Here is a precise unranking construction. Fix an order of the branches in every sum. Given a rank  $q$  between zero and the count minus one, locate its branch by subtracting preceding branch weights. For a branch combining two intervals with counts  $a, b$  and lengths  $s, t$ , its weight is

$$w = ab \binom{s+t}{s}.$$

Every  $0 \leq q < w$  has a unique representation

$$q = q_L + aq_R + abq_{\text{sh}}, \quad 0 \leq q_L < a, \quad 0 \leq q_R < b, \quad 0 \leq q_{\text{sh}} < \binom{s+t}{s}.$$

Recursively unrank the two interval orders and then their shuffle. To unrank a shuffle, list those starting with a left symbol first. There are  $\binom{s+t-1}{s-1}$  such shuffles. A comparison and subtraction therefore determine its first symbol, and iteration determines the whole word.

In (4), append the chosen last vertex to the shuffle. In (5) and (8), leave the selected vertex or pair unpressed. In (6), append the selected vertex after unranking the complementary path; in (7), do not append it. Disjoint components are combined successively by the same mixed-radix decomposition and shuffle rule. Empty intervals have their unique empty order. Zero-weight branches are skipped.

These rules give a bijection between the integers below the total count and the counted sequences, by induction on interval length and then on the number of components. Interval branch scans and shuffle construction cost at most  $O(n^2)$  arithmetic operations per unranking: at each recursion node the cost is linear in its interval length, and the sum of subtree sizes is at most  $n^2$ . The initial cycle pair scan also costs at most  $O(n^2)$ . Component combinations obey the same bound. A single exact uniform integer  $q$  therefore yields an exact uniform sequence. With fair random bits, ordinary rejection from a power-of-two range uses fewer than two trials on average and  $O(1 + n \log(n+1))$  expected bits. A count of one requires no random bits. This completes the proof of Theorem 1.

## 6 Reproducibility and scope

The accompanying standard-library Python implementation compares the recurrence with a separate forward simulation of closed-neighborhood complementation. It checks all 8,191 colorings of paths of orders zero through twelve, all 2,040 colorings of cycles of orders three through ten, and 300 seeded disjoint-union instances. Both counting conventions are checked, giving 21,062 comparisons. Exhaustive unranking on the path and cycle instances through order seven validates 38,426 sequences, checking legality, success, and absence of duplicates. As a further consistency check, full counts summed over all colorings equal  $n!$ , in agreement with [1, Theorem 13]. These finite checks support the proof and are not its substitute.

The interval recurrence, the rank-deficiency cases, and exact unranking form a complete result for the stated class. The general-graph hardness and approximate-counting questions in AMR-030-0046 remain unresolved here. Exact sampling on paths also does not establish a mixing-time bound for a particular local-edit Markov chain. A bounded primary-source review did not locate the same path-and-cycle counting algorithm; no exhaustive priority claim is made. AI assistance was used in developing the argument, searching sources, writing code, and preparing this manuscript. The argument was checked within that workflow; this preprint has not undergone independent peer review or proof-assistant verification.

## References

- [1] J. Cooper and J. Davis, Successful pressing sequences for a bicolored graph and binary matrices, *Linear Algebra Appl.* **490** (2016), 162–173. doi:[10.1016/j.laa.2015.11.001](https://doi.org/10.1016/j.laa.2015.11.001).
- [2] E. Bixby, T. Flint, and I. Miklós, Proving the pressing game conjecture on linear graphs, *Involve* **9** (2016), no. 1, 41–56. doi:[10.2140/involve.2016.9.41](https://doi.org/10.2140/involve.2016.9.41).
- [3] D. Bevan, M. Bóna, and I. Miklós (eds.), Pattern Avoidance, Statistical Mechanics and Computational Complexity (Dagstuhl Seminar 23121), *Dagstuhl Reports* **13** (2023), no. 3, 49–73. I. Miklós, Section 4.8, p. 71. doi:[10.4230/DagRep.13.3.49](https://doi.org/10.4230/DagRep.13.3.49).
- [4] J. N. Cooper and H. W. Whitlatch, Uniquely pressable graphs: characterization, enumeration, and recognition, *Adv. Appl. Math.* **103** (2019), 13–42. doi:[10.1016/j.aam.2018.09.005](https://doi.org/10.1016/j.aam.2018.09.005).
- [5] J. Cooper, P. Gartland, and H. Whitlatch, A new characterization of  $\mathcal{V}$ -posets, arXiv:1810.07276v2 (2018). <https://arxiv.org/abs/1810.07276>.
- [6] L. Bailey et al., Complexity and enumeration in models of genome rearrangement, *Theoret. Comput. Sci.* **1022** (2024), article 114880. doi:[10.1016/j.tcs.2024.114880](https://doi.org/10.1016/j.tcs.2024.114880).
- [7] J. Cooper, Combinatorial Problems I Like, October 2020 snapshot; consulted 25 October 2026. <https://people.math.sc.edu/cooper/combprob.html>.
- [8] ulamai, *UnsolvedMath*, version 1.6.0, record AMR-030-0046, frozen commit c6e7f41b4eb6ce876feb83e7eb626be1b243fd7e. <https://huggingface.co/datasets/ulamai/UnsolvedMath>.