WIKIPEDIA

Stack-sortable permutation

In mathematics and computer science, a **stack-sortable permutation** (also called a **tree permutation**)[1] is a permutation whose elements may be sorted by an algorithm whose internal storage is limited to a single stack data structure. The stack-sortable permutations are exactly the permutations that do not contain the permutation pattern 231; they are counted by the Catalan numbers, and may be placed in bijection with many other combinatorial objects with the same counting function including Dyck paths and binary trees.

Contents

Sorting with a stack

The problem of sorting an input sequence using a stack was first posed by Knuth (1968), who gave the following linear time algorithm (closely related to algorithms for the later all nearest smaller values problem):

- Initialize an empty stack
- For each input value *x*:
  - While the stack is nonempty and *x* is larger than the top item on the stack, pop the stack to the output
  - Push *x* onto the stack
- While the stack is nonempty, pop it to the output

Knuth observed that this algorithm correctly sorts some input sequences, and fails to sort others. For instance, the sequence 3,2,1 is correctly sorted: the three elements are all pushed onto the stack, and then popped in the order 1,2,3. However, the sequence 2,3,1 is not correctly sorted: the algorithm first pushes 2, and pops it when it sees the larger input value 3, causing 2 to be output before 1 rather than after it.

Because this algorithm is a comparison sort, its success or failure does not depend on the numerical values of the input sequence, but only on their relative order; that is, an input may be described by the permutation needed to form that input from a sorted sequence of the same length. Knuth characterized the permutations that this algorithm correctly sorts as being exactly the permutations that do not contain the permutation pattern 231: three elements x, y, and z, appearing in the input in that respective order, with $z < x < y$. Moreover, he observed that, if the algorithm fails to sort an input, then that input cannot be sorted with a single stack.

As well as inspiring much subsequent work on sorting using more complicated systems of stacks and related data structures,[2] Knuth's research kicked off the study of permutation patterns and of permutation classes defined by forbidden patterns.

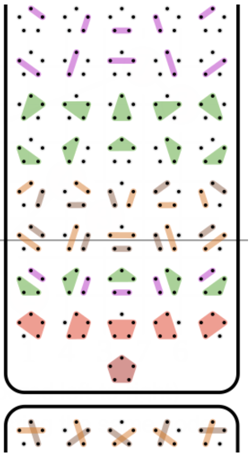
Bijections and enumeration

The sequence of pushes and pops performed by Knuth's sor
reinterpreting a push as a left parenthesis and a pop as a right p
comes from a stack-sortable permutation in this way, and every
reason, the number of stack-sortable permutations of length *n* is th

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$[3]

Stack-sortable permutations may also be translated directly to an from (unlabeled) binary trees, another combinatorial class whos counting function is the sequence of Catalan numbers. A binar tree may be transformed into a stack-sortable permutation b numbering its nodes in left-to-right order, and then listing thes numbers in the order they would be visited by a preorder traversa of the tree: the root first, then the left subtree, then the righ subtree, continuing recursively within each subtree. In the revers direction, a stack-sortable permutation may be decoded into a tre in which the first value x of the permutation corresponds to th root of the tree, the next $x - 1$ values are decoded recursively t give the left child of the root, and the remaining values are agai decoded recursively to give the right child.[1]

In combinatorial mathematics, the **Catalan numbers** form a sequence of natural numbers that occur in various counting problems, often involving recursively-defined objects.



numbers in preorder

Several other classes of permutations may also be placed in bijection with the stack-sortable permutations. For instance, the permutations that avoid the patterns 132, 213, and 312 may be formed respectively from the stack-sortable (231-avoiding) permutations by reversing the permutation, replacing each value x in the permutation by $n + 1 - x$, or both operations combined. The 312-avoiding permutations are also the inverses of the 231-avoiding permutations, and have been called the **stack-realizable permutations** as they are the permutations that can be formed from the identity permutation by a sequence of push-from-input and pop-to-output operations on a stack.[4] As Knuth (1968) noted, the 123-avoiding and 321-avoiding permutations also have the same counting function despite being less directly related to the stack-sortable permutations.

Random stack-sortable permutations

Rotem (1981) investigates the properties of stack-sortable permutations chosen uniformly at random among all such permutations of a given length. The expected length of the longest descending subsequence in such a permutation is $\sqrt{\pi n} - O(1)$, differing by a constant factor from unconstrained random permutations (for which the expected length is approximately $2\sqrt{n}$). The expected length of the longest ascending sequence differs even more strongly from unconstrained permutations: it is $(n+1)/2$. The expected number of values within the permutation that are larger than all previous values is only $3 - 6/(n+2)$, smaller than its logarithmic value for unconstrained permutations. And the expected number of inversions is $\Theta(n^{3/2})$, in contrast to its value of $\Theta(n^2)$ for unconstrained permutations.

Additional properties

Every permutation defines a permutation graph, a graph whose vertices are the elements of the permutation and whose edges connect pairs of elements that are inverted by the permutation. The permutation graphs of stack-sortable graphs are trivially perfect.[4]

For each element i of a permutation p, define b_i to be the number of other elements that are to the left of and greater than i. Then p is stack-sortable if and only if, for all i, $b_i - b_{i+1} \leq 1$.[1]

Algorithms

Knott (1977) uses the bijection between stack-sortable permutations and binary trees to define a numerical rank for each binary tree, and to construct efficient algorithms for computing the rank of a tree ("ranking") and for computing the tree with a given rank ("unranking").

Micheli & Rossin (2006) defined two edit operations on permutations: deletion (making a permutation pattern) and its inverse. Using the same correspondence between trees and permutations, they observed that these operations correspond to edge contraction in a tree and its inverse. By applying a polynomial time dynamic programming algorithm for edit distance in trees, they showed that the edit distance between two stack-sortable permutations (and hence also the longest common pattern) can be found in polynomial time. This technique was later generalized to algorithms for finding longest common patterns of separable permutations;[5] however, the longest common pattern problem is NP-complete for arbitrary permutations.[6]

Notes

1. Knott (1977).
2. Tarjan (1972); Avis & Newborn (1981); Rosenstiehl & Tarjan (1984); Bóna (2002); Felsner & Pergel (2008). See also the many additional references given by Bóna.
3. Knuth (1968); Rotem (1981).

4. Rotem (1981).
5. Bouvel, Rossin & Vialette (2007).
6. Micheli & Rossin (2006).

References

- Avis, David; Newborn, Monroe (1981), "On pop-stacks in series", *Utilitas Mathematica*, **19**: 129–140, MR 0624050 (https://www.ams.org/mathscinet-getitem?mr=0624050).
- Bóna, Miklós (2002), "A survey of stack-sorting disciplines" (http://www.combinatorics.org/Volume_9/Abstracts/v9i2a1.html), *Electronic Journal of Combinatorics*, **9** (2): A1, MR 2028290 (https://www.ams.org/mathscinet-getitem?mr=2028290).
- Bouvel, Mathilde; Rossin, Dominique; Vialette, Stéphane (2007), "Longest common separable pattern among permutations", *Combinatorial Pattern Matching (CPM 2007)*, Lecture Notes in Computer Science, **4580**, Springer, pp. 316–327, doi:10.1007/978-3-540-73437-6_32 (https://doi.org/10.1007%2F978-3-540-73437-6_32).
- Felsner, Stefan; Pergel, Martin (2008), "The complexity of sorting with networks of stacks and queues", *Proc. 16th Eur. Symp. Algorithms*, Karlsruhe, Germany, pp. 417–429, ISBN 978-3-540-87743-1, doi:10.1007/978-3-540-87744-8_35 (https://doi.org/10.1007%2F978-3-540-87744-8_35).
- Knott, Gary D. (February 1977), "A numbering system for binary trees", *Communications of the ACM*, **20** (2): 113–115, doi:10.1145/359423.359434 (https://doi.org/10.1145%2F359423.359434).
- Knuth, Donald (1968), "Vol. 1: Fundamental Algorithms", *The Art of Computer Programming*, Reading, Mass.: Addison-Wesley.
- Micheli, Anne; Rossin, Dominique (2006), "Edit distance between unlabeled ordered trees", *Theoretical Informatics and Applications*, **40** (4): 593–609, MR 2277052 (https://www.ams.org/mathscinet-getitem?mr=2277052), arXiv:math/0506538 (https://arxiv.org/abs/math/0506538), doi:10.1051/ita:2006043 (https://doi.org/10.1051%2Fita%3A2006043).
- Rosenstiehl, Pierre; Tarjan, Robert E. (1984), "Gauss codes, planar Hamiltonian graphs, and stack-sortable permutations", *Journal of Algorithms*, **5** (3): 375–390, MR 756164 (https://www.ams.org/mathscinet-getitem?mr=756164), doi:10.1016/0196-6774(84)90018-X (https://doi.org/10.1016%2F0196-6774%2884%2990018-X)
- Rotem, D. (1981), "Stack sortable permutations", *Discrete Mathematics*, **33** (2): 185–196, MR 599081 (https://www.ams.org/mathscinet-getitem?mr=599081), doi:10.1016/0012-365X(81)90165-5 (https://doi.org/10.1016%2F0012-365X%2881%2990165-5).
- Tarjan, Robert (April 1972), "Sorting Using Networks of Queues and Stacks", *Journal of the ACM*, **19** (2): 341–346, doi:10.1145/321694.321704 (https://doi.org/10.1145%2F321694.321704).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Stack-sortable_permutation&oldid=711280898"

This page was last edited on 2016-03-22, at 07:04:18.