Planning a Gift Exchange

Darryl K. Nester
Professor of Mathematics
Bluffton University

Abstract: In a simple gift exchange ("Secret Santa"), each person in a group chooses the name of someone else, for whom the first person will buy a gift. A collection of such matches—where no one has his or her own name—is called a "derangement." One might wish to impose additional restrictions; for example, in my extended family's annual Christmas exchange, we decided that no one should choose the name of his/her spouse.

The low-tech way to do this is for each person to pick a name from a hat, repeating the process until all participants have a suitable name. However, you can use any of a number of Web sites to automate the selection, including the one I created for my family's exchange (link below).

We'll count the number of derangements in an unrestricted gift exchange, and then explore how that number changes when we make additional restrictions, using a counting technique called "rook polynomials." In addition, we'll discuss the practical issues in implementing a gift exchange Web page, such as keeping the matches secret from the person organizing the exchange.

http://www.bluffton.edu/~nesterd/java/secretsanta.html

2015 Miami University Mathematics Conference
Combinatorics and its Applications

Note: This document was originally formatted for use in a presentation (landscape, with a larger font). For its new role as a supplemental technical report (with expanded content), it has been reformatted.

For simplicity, we can model an exchange among (say) 5 people as a list of numbers indicating the recipient chosen for each person. As computers typically do, we will call the 5 participants 0, 1, 2, 3, 4 (rather than 1…5), so a possible gift exchange might be $\{4, 2, 1, 0, 3\}$ (hereafter written as 42103), which means that 0 buys for 4, 1 buys for 2, etc.

Permuting (shuffling) a list can be done using a Fisher-Yates (or Knuth) shuffle, which is essentially equivalent to shuffling a deck of cards by choosing a card at random from the whole deck to be "first," then picking the second card from the remaining cards, then the third card, etc.

But we need a permutation of $\{0, 1, 2, \dots, n-1\}$ in which all the numbers are displaced; that is, we don't want k to be in position k. Such a permutation is called a *derangement*.

We could just shuffle the list and then check if it is a derangement. It happens that the fraction of the $n!$ permutations which are derangements is about $1/e$, so on average we'd need to generate e permutations to find each derangement … and that might not satisfy the other requirements of our exchange, so we'd need to start again!

So instead we use a more efficient way to generate derangements from the paper:

> Martínez, Panholzer, and Prodinger, "Generating Random Derangements." 2008 Proceedings of the Fifth Workshop on Analytic Algorithmics and Combinatorics (ANALCO), pp. 234-240.
> http://epubs.siam.org/doi/pdf/10.1137/1.9781611972986.7

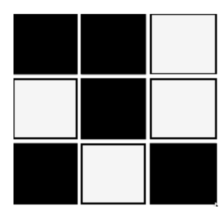
This saves a bit of time (though we still have to check the other restrictions on the exchange).

For a given set of restrictions, how many possible assignments can be made?

To count the valid assignments, we use "rook polynomials," which can be used to count the number of ways to put rooks on selected squares of a chessboard in such a way that no two are in the same row or column (i.e., "non-capturing rooks"). This happens to be equivalent to the number of permutations satisfying a given set of restrictions.

Consider the 3×3 chessboard on the right, where we can place rooks on the black squares. The rook polynomial for this board is

$$r_1(x) = 1 + 5x + 6x^2 + 1x^3$$

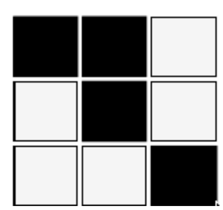


because we can place:

- 0 rooks in 1 way
- 1 rook in 5 ways
- 2 rooks in 6 ways
- 3 rooks in 1 way

By contrast, this second chessboard has rook polynomial

$$r_2(x) = 1 + 4x + 4x^2 + 1x^3$$



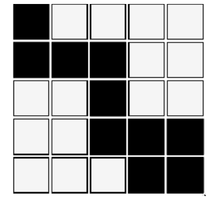
Note that (not too surprisingly) with one less place to put a rook, there are fewer ways to place 1 or 2 rooks on the board.

It is fairly simple to write a computer program to find the rook polynomial for any such chessboard.

You can explore rook polynomials further at

http://www.bluffton.edu/~nesterd/java/rookpolynomials.html

How is the non-capturing rooks problem related to the restricted permutations problem?



The black squares on the chessboard correspond to the disallowed selections in the gift exchange. For example, the chessboard on the right would represent a derangement (because the main diagonal is black), for which persons 0 and 2 can choose anyone else, person 1 can choose 3 or 4, person 3 can choose 0 or 1, and person 4 can choose 0, 1, or 2.

The rook polynomial for this board is

$$1 + 10x + 32x^2 + 38x^3 + 16x^4 + 2x^5$$

and the number of valid derangements is

$$1\times5! - 10\times4! + 32\times3! - 38\times2! + 16\times1! - 2\times0! = 10$$

Categorizing permutations

In order to save a set of assignments, we need a convenient way to encode the permutation. The approach I took was "lexicographic order"—that is, we assign to each of the $n!$ permutations a number (index) from 0 to $n! - 1$, reflecting its position in the natural ("alphabetical") order of permutations. For example, with $n = 3$:

Permutation index #	Corresponding permutation	
0	012	
1	021	
2	102	
3	120	(a derangement)
4	201	(a derangement)
5	210	

As two more examples: For $n = 4$, permutation #17 is 2310, and for $n = 5$, permutation #93 is 34120.

This is hardly a unique approach, and I don't claim to be the first one to take it. A quick Google search for "permutation index" or "permutation lexicographic order" brings up other resources; e.g., https://en.wikipedia.org/wiki/Permutation#Numbering_permutations. Nevertheless, for completeness, we'll address the obvious questions here:

(1) How do you translate a given index number into a permutation?
(2) How do you translate a given permutation into an index number?

To answer the first question, consider some special cases. In the list of permutations above (for $n = 3$), note that it divides neatly into thirds: The first two begin with 0, the next two with 1, and the last two with 2. For $n = 4$, if we list the 24 permutations of {0,1,2,3} in order, note that:

- permutations 0–5 (the first six) begin with 0,
- permutations 6–11 begin with 1,
- permutations 12–17 begin with 2, and
- permutations 18–23 begin with 3.

Likewise, for $n = 5$:

- permutations 0–23 (the first 24) begin with 0,
- permutations 24–47 begin with 1,
- permutations 48–71 begin with 2,
- permutations 72–95 begin with 3, and
- permutations 96–119 begin with 4.

In general, for a given number of people n and index number k_0, compute

$$j_1 = \left\lfloor \frac{k_0}{(n-1)!} \right\rfloor \quad \text{and} \quad k_1 = k_0 \bmod (n-1)! = k_0 - j_1 \times (n-1)!$$

The permutation begins with j_1, and the remaining $n - 1$ items

$$\{0, 1, 2, 3, \dots, n\} - \{ j_1 \} = \{0, 1, 2, \dots, j_1 - 1, j_1 + 1, \dots, n\}$$

are arranged in the order determined by the permutation index k_1 (with the understanding that $j_1 + 1$ now takes the place of j_1, etc.).

For example, the 94th permutation of 01234 (which has index $k_0 = 93$) is 34120, because:

n	$(n-1)!$	j_i	k_i	Comment
5	24	$\left\lfloor \frac{93}{24} \right\rfloor = 3$	$93 \bmod 24 = 21$	Permutation starts with 3
4	6	$\left\lfloor \frac{21}{6} \right\rfloor = 3$	$21 \bmod 6 = 3$	... then item 3 from 0124
3	2	$\left\lfloor \frac{3}{2} \right\rfloor = 1$	$3 \bmod 2 = 1$	... then item 1 from 012
2	1	$\left\lfloor \frac{1}{1} \right\rfloor = 1$	$1 \bmod 1 = 0$	... then item 1 from 02
1	1	0	—	... then item 0 from 0.

Likewise, the 198th permutation ($k_0 = 197$) of 012345 is 140532:

n	$(n-1)!$	j_i	k_i	Comment
6	120	1	77	Permutation starts with 1
5	24	3	5	... then item 3 from 02345
4	6	0	5	... then item 0 from 0235
3	2	2	1	... then item 2 from 235
2	1	1	0	... then item 1 from 23
1	1	0	—	... then item 0 from 2.

And the 501st permutation of ABCDEF is EAFCBD:

n	$(n-1)!$	j_i	k_i	Comment
6	120	4	20	Permutation starts with E
5	24	0	20	... then item 0 from ABCDF
4	6	3	2	... then item 3 from BCDF
3	2	1	0	... then item 1 from BCD
2	1	0	0	... then item 0 from BD
1	1	0	—	... then item 0 from D.

The inverse question: Given a permutation, what is its index?

It is fairly straightforward to reverse the process illustrated above. We repeatedly remove the first item in the permutation, and adjusting its value based on the other remaining items. For example, if after we have removed several items, the remaining permutation is 3154, the value of 3 is adjusted to 1 (because it is the second-smallest of the remaining numbers). For the partial permutation 62304, the value of 6 would be adjusted to 4 (the largest of 5 items).

The index is the sum of each of these values, multiplied by an appropriate factorial (namely, one less than the length of the permutation). For example, for the permutation 304512:

Remaining Permutation	Value of first item	Term
304512	3	$3 \times 5!$ = 360
04512	0	$0 \times 4!$ = 0
4512	2	$2 \times 3!$ = 12
512	2	$2 \times 2!$ = 4
12	0	$0 \times 1!$ = 0
2	0	$0 \times 0!$ = 0

So the index would be 376. Likewise, the permutation FCEGADB has index 4651:

Remaining Permutation	Value of first item	Term
FCEGADB	6	$6 \times 6!$ = 4320
CEGADB	2	$2 \times 5!$ = 240
EGADB	3	$3 \times 4!$ = 72
GADB	3	$3 \times 3!$ = 18
ADB	0	$0 \times 2!$ = 0
DB	1	$1 \times 1!$ = 1
B	0	$0 \times 0!$ = 0