Fixed points of permutations

Let $f : S \to S$ be a permutation of a set S. An element $s \in S$ is a **fixed point** of f if $f(s) = s$. That is, the fixed points of a permutation are the points *not moved* by the permutation.

For example,

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 2 & 6 & 5 & 4 \end{pmatrix}$$

has fixed points $\{1, 5\}$, since $f(1) = 1$ and $f(5) = 5$ and everything else is sent to something different.

We might be interested in whether a *random* permutation has fixed points, or not, or how many we should expect it to have, and such things.

This is a good exercise in counting, as well as informative about random choices of *mixing* functions.

Thinking of a (block) cipher as a permutation (depending on the key) on strings of a certain size, we would *not* want such a permutation to have many fixed points.

Information about *typical* behavior of permutations may shed light on how hard we might expect to have to work to achieve a whole *family* of good mixing effects, parametrized by the *key*.

In **symmetric ciphers** such as DES, AES (Rijndael), as opposed to **asymmetric (public-key) ciphers** such as RSA, the whole cipher is usually put together from smaller pieces (**S-boxes**) that do the critical and hopefully very tricky mixing.

To count permutations of $\{1, \ldots, 10\}$ **having at least one fixed point**:
at least 3 approaches: an **inclusion-exclusion** approach (maybe most intuitive), a **recursive** approach (slicked-up version of inclusion-exclusion), and a **cycle-structure** approach with the virtue that it gives a sort of formula, though not so useful for numerical evaluation.

Try to count permutations having at least one fixed point

$$\text{no. fixing } \texttt{`1'} + \text{no. fixing } \texttt{`2'}$$

$$+\text{no. fixing } \texttt{`3'} + \ldots + \text{no. fixing } \texttt{`10'}$$

$$= \binom{10}{1} \cdot (10-1)!$$

since there are $\binom{10}{1}$ choices of single-element subset to be fixed, and for each choice there are $(10-1)!$ permutations altogether of the remaining $10-1$ elements.

But this definitely **overcounts**: a permutation that fixes more than one element occurs in more than one of the summands.

Try to compensate by *subtracting* from the previous count the quantity

$$\text{no. fixing } \texttt{`1'} \textit{ and } \texttt{`2'}$$

$$+ \text{ no. fixing } \texttt{`1'} \textit{ and } \texttt{`3'}$$

$$+ \ldots + \text{ no. fixing } \texttt{`1'} \textit{ and } \texttt{`10'}$$

$$+ \text{ no. fixing } \texttt{`2'} \textit{ and } \texttt{`3'}$$

$$+ \ldots + \text{ no. fixing } \texttt{`9'} \textit{ and } \texttt{`10'}$$

$$= \binom{10}{2} \cdot (10-2)!$$

with $\binom{10}{2}$ choices of two-element subset to be fixed, and for each choice $(10-2)!$ permutations of the remaining $10-2$ elements.

So far we've *approximated* the number of permutations with at least one fixed point as

$$\binom{10}{1} \cdot (10-1)! - \binom{10}{2} \cdot (10-2)!$$

But now we've have over-counted or under-counted permutations fixing at least 3 elements.

Indeed, if a permutation P fixes exactly 3 elements it will have been counted $\binom{3}{1}$ times in the first summand in that last expression, once for each 1-element subset of the 3 elements, and $\binom{3}{2}$ times in the second summand, once for each 2-element subset of the 3 elements. Thus, the *net* count so far of such a permutation is

$$\binom{3}{1} - \binom{3}{2} = 3 - 3 = 0$$

But we want the net count to be 1.

To compensate for this miscount we *add*

$$\text{no. perms fixing } \texttt{1,2,3}$$

$$+ \text{ no. perms fixing } \texttt{1,2,4}$$

$$+ \ldots + \text{ no. perms fixing } \texttt{8,9,10}$$

$$= \binom{10}{3} \cdot (10-3)!$$

Thus, so far, the attempted count would be

$$\binom{10}{1} \cdot (10-1)! - \binom{10}{2} \cdot (10-2)!$$

$$+ \binom{10}{3} \cdot (10-3)!$$

The *net count* of permutations fixing exactly 4 things so far is

$$\binom{4}{1} - \binom{4}{2} + \binom{4}{3} = 4 - 6 + 4 = 2$$

So we've *overcounted by 1* permutations fixing 4 elements so far, so *subtract*

$$\text{no. fixing } \texttt{1,2,3,4}$$

$$+ \text{ no. fixing } \texttt{1,2,3,5}$$

$$+ \ldots$$

$$+ \text{ no. fixing } \texttt{7,8,9,10}$$

$$= \binom{10}{4} \cdot (10-4)!$$

Net count of permutations fixing exactly 5 things:

$$\binom{5}{1} - \binom{5}{2} + \binom{5}{3} - \binom{5}{4} = 5 - 10 + 10 - 5 = 0$$

We've *undercounted by 1* permutations fixing 5 so far, so *add*

$$\text{no. fixing } \texttt{1,2,3,4,5}$$

$$+ \text{ no. fixing } \texttt{1,2,3,4,6}$$

$$+ \dots$$

$$+ \text{ no. fixing } \texttt{6,7,8,9,10}$$

$$= \binom{10}{5} \cdot (10-5)!$$

The net count of permutations fixing exactly 6 things: it would be

$$\binom{6}{1} - \binom{6}{2} + \binom{6}{3} - \binom{6}{4} + \binom{6}{5}$$

$$= 6 - 15 + 20 - 15 + 6 = 2$$

So we've *overcounted by 1* so far,

so *subtract*

$$\text{no. perms fixing } 1,2,3,4,5,6$$

$$+\ldots+ \text{ no. perms fixing } 5,6,7,8,9,10$$

$$= \binom{10}{6} \cdot (10-6)!$$

Look at the net count of permutations fixing exactly 7 things: it would be

$$\binom{7}{1} - \binom{7}{2} + \binom{7}{3} - \binom{7}{4} + \binom{7}{5} - \binom{7}{6} = 0$$

So we've undercounted by 1 so far, so *add*

$$\text{no. perms fixing } 1,2,3,4,5,6,7$$

$$+\ldots+ \text{ no. perms fixing } 4,5,6,7,8,9,10$$

$$= \binom{10}{7} \cdot (10-7)!$$

The net count of permutations fixing exactly 8 things so far is

$$\binom{8}{1} - \binom{8}{2} + \binom{8}{3} - \binom{8}{4} + \binom{8}{5} - \binom{8}{6} + \binom{8}{7}$$

$$= 8 - 28 + 56 - 70 + 56 - 28 + 8 = 2$$

(Has anyone started wondering why we've been so lucky that we've always either over-counted or under-counted by 1, and in alternating cases?)

We've overcounted by 1 so far, so *subtract*

$$\text{no. fixing } \texttt{1,2,3,4,5,6,7,8}$$

$$+ \ldots + \text{ no. fixing } \texttt{3,4,5,6,7,8,9,10}$$

$$= \binom{10}{8} \cdot (10 - 8)!$$

The net count of permutations fixing exactly 9 things is would be

$$\binom{9}{1} - \binom{9}{2} + \binom{9}{3} - \binom{9}{4} + \ldots + \binom{9}{7} - \binom{9}{8} = 0$$

(For odd k such as $k = 9$, as in the odd case, we can use the fact that $\binom{k}{i} = \binom{k}{k-i}$ and the opposite signs that occur in the net count expression to see that we'll get a net count of 0*, but why do we always get a net count of 2 in the even case?)*

We've undercounted by 1 so far, so *add*

$$\text{no. fixing } \texttt{1,2,3,4,5,6,7,8,9}$$

$$+ \ldots + \text{ no. fixing } \texttt{2,3,4,5,6,7,8,9,10}$$

$$= \binom{10}{9} \cdot (10-9)!$$

The net count of permutations fixing exactly 10 things is

$$\binom{10}{1} - \binom{10}{2} + \binom{10}{3} - \binom{10}{4} + \binom{10}{5}$$

$$-\binom{10}{6} + \binom{10}{7} - \binom{10}{8} + \binom{10}{9}$$

$$= 10 - 45 + 120 - 210 + 252 - 210 + 120 - 45 + 10$$

$$= 2$$

We've overcounted by 1 so far, so *subtract*

$$\text{no. perms fixing } 1,2,3,4,5,6,7,8,9,10$$

$$= \binom{10}{10} \cdot (10-10)! = 1$$

Thus, in summary, the number of permutations of 10 things fixing at least one element is

$$\binom{10}{1}(10-1)! - \binom{10}{2}(10-2)!$$

$$+\binom{10}{3}(10-3)! - \binom{10}{4}(10-4)!$$

$$+\binom{10}{5}(10-5)! - \binom{10}{6}(10-6)!$$

$$+\binom{10}{7}(10-7)! - \binom{10}{8}(10-8)!$$

$$+\binom{10}{9}(10-9)! - \binom{10}{10}(10-10)!$$

How to evaluate this nicely? Not clear yet.

And what about that little point about why we were so lucky as to be off by only ± 1 in the net count?

The Binomial Theorem asserts

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i\, y^{n-i}$$

In particular, with $x = 1$ and $y = -1$,

$$0 = (1-1)^n$$

$$= 1 - \sum_{k=1}^{n-1} (-1)^k \binom{n}{k} + (-1)^n$$

Rearrange to

$$\sum_{k=1}^{n-1} (-1)^k \binom{n}{k} = 1 + (-1)^n = \begin{cases} 2 & (n \text{ even}) \\ 0 & (n \text{ odd}) \end{cases}$$

Recursive approach

Let $f(n)$ be the number of permutations of n things with *no* fixed point.

And

$$\text{no. perms of } n \text{ fixing at least one}$$

$$= \sum_{k=1}^{n} (\text{no. perms fixing exactly } k \text{ elts})$$

$$= \sum_{k=1}^{n} \binom{n}{k} \cdot f(n-k)$$

since there are $\binom{n}{k}$ k-element subsets of n things to choose as the exact fixed-point set, and $f(n-k)$ counts the number of permutations of the remaining $n-k$ which do move every one.

Then

$$\text{no. perms of } n \text{ fixing at least one}$$

$$= \text{no. all perms of } n \text{ things}$$

$$-\text{no. perms of } n \text{ things fixing none}$$

$$= n! - f(n)$$

Sticking these two relations together, we get the recursive relation

$$f(n) = n! - \sum_{k=1}^{n} \binom{n}{k} \cdot f(n-k)$$

which expresses each $f(n)$ in terms of $f(\ell)$ with $\ell < n$.

Note that this requires the perhaps-surprising convention that $f(0) = 1$.

Thus, counting the number of permutations of n things with no fixed points, for $n = 0, 1, 2, \ldots$:

$$
\begin{aligned}
f(0) &= \mathbf{1} \\
f(1) &= 1! - \binom{1}{1} \cdot f(0) = 1 - 1 = \mathbf{0} \\
f(2) &= 2! - \binom{2}{1} \cdot f(1) - \binom{2}{2} \cdot f(0) \\
&= 2 - 2 \cdot 0 - 1 \cdot 1 = \mathbf{1} \\
f(3) &= 3! - \binom{3}{1} f(2) - \binom{3}{2} f(1) - \binom{3}{3} f(0) \\
&\quad = 6 - 3 \cdot 1 - 3 \cdot 0 - 1 = \mathbf{2} \\
f(4) &= 4! - \binom{4}{1} \cdot f(3) - \binom{4}{2} \cdot f(2) \\
&\qquad - \binom{4}{3} \cdot f(1) - \binom{4}{4} \cdot f(0) \\
&= 24 - 4 \cdot 2 - 6 \cdot 1 - 4 \cdot 0 - 1 = \mathbf{9} \\
f(5) &= 5! - \binom{5}{1} f(4) - \binom{5}{2} f(3) \\
&\qquad - \binom{5}{3} f(2) - \binom{5}{4} f(1) - \binom{5}{5} f(0) \\
&= 120 - 5 \cdot 9 - 10 \cdot 2 - 10 \cdot 1 - 0 - 1 \\
&\quad = \mathbf{44} \\
f(6) &= 6! - \binom{6}{1} f(5) - \binom{6}{2} f(4) \\
&\quad - \binom{6}{3} f(3) - \binom{6}{4} f(2) - \binom{6}{5} f(1) - 1 \\
&= 720 - 6 \cdot 44 - 15 \cdot 9 - 20 \cdot 2 \\
&\qquad -15 \cdot 1 - 0 - 1 = \mathbf{265}
\end{aligned}
$$

This is no picnic for large values of n.

Cycle-structure approach

We can determine the number $f(n)$ of permutations of n things *without fixed points* in another way, by counting the possible disjoint-cycle decompositions that would give such a permutation.

That is, we count the number of products of disjoint cycles such that every element of the set $\{1, \ldots, n\}$ occurs in some cycle of length 2 or more.

That is, we sum over $2 \leq k_1 \leq k_2 \leq \ldots, k_t$ with variable t and with

$$k_1 + k_2 + \ldots + k_t = n$$

and count the number of products of disjoint k_1-cycle, k_2-cycle, ..., k_t-cycles.

For very large n this is again not feasible, but...

To compute $f(5)$:

Since $2 \leq k_i$ with k_1 at its smallest possible value $k_1 = 2$, k_2 can be either 2 or 3, but must be $k_2 = 3$ because of the condition $\sum_i k_i = 5$. (There is no room for a k_3 in any case.) Thus, we have products of disjoint 2-cycles and 3-cycles.

The number of disjoint products of 2-cycles and 3-cycles is

$$\frac{5 \cdot 4}{2} \cdot \frac{3 \cdot 2 \cdot 1}{3} = 20$$

because we have 5 choices for the first element in the 2-cycle, then 4 choices for the second, but then must divide by 2 since there are two ways to write the same 2-cycle. Similarly, for each such choice there are 3 choices for the first element of the 3 cycle, 2 for the second, and 1 for the third, but divide by 3 because each 3-cycle can be written 3 ways.

If $k_1 > 2$ then there is no room for any more k_is and we conclude that $k_1 = 5$. And indeed 5 cycles have no fixed points.

The number of 5-cycles is

$$\frac{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{5} = 24$$

since we have 5 choices for first element, etc., but divide by 5 since each 5 cycle can be written 5 ways.

Altogether there are

$$f(5) = \text{no. disjoint 3-cycles and 2-cycles}$$

$$+\text{no. 5-cycles}$$

$$= 20 + 24 = 44$$

matching the recursive result.

For $f(6)$:

The possible sets of cycle lengths are 2,2,2 and 2,4 and 3,3 and 6, obtained as follows, by looking down a list of candidates in a sort of recursive lexicographic order.

For the smallest value $k_1 = 2$, we have $2 \leq k_2 \leq \ldots$ and $k_2 + \ldots = 4$. With the smallest value $k_2 = 2$, there is only one choice $k_3 = 2$. With $k_2 = 3$ we fail. With $k_2 = 4$ we again succeed.

With $k_1 = 3$, $3 \leq k_2$, leavning one choice $k_2 = 3$.

Values $k_1 = 4, 5$ fail since we cannot hit the sum 6, but $k_1 = 6$ is ok by itself.

The number of disjoint products of 2-cycle, 2-cycle, 2-cycle is

$$\frac{6 \cdot 5}{2} \cdot \frac{4 \cdot 3}{2} \cdot \frac{2 \cdot 1}{2} \cdot \frac{1}{3!} = 15$$

Divide by 3! since we will have chosen the same *permutation* 3! different ways: disjoint cycles can be written in any order. (They **commute**.)

Disjoint products of 2-cycle, 4-cycle is

$$\frac{6 \cdot 5}{2} \cdot \frac{4 \cdot 3 \cdot 2 \cdot 1}{4} = 90$$

Disjoint products of 3-cycle, 3-cycle is

$$\frac{6 \cdot 5 \cdot 4}{3} \cdot \frac{3 \cdot 2 \cdot 1}{3} \cdot \frac{1}{2!} = 40$$

And 6-cycles

$$\frac{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{6} = 120$$

$$Total = 15 + 90 + 40 + 120 = 265 \textit{(matches!)}$$

Approximation for large n

Ironically, the first approach gives an *approximate* value for large n.

$$f(n) = n! - \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!$$

$$= n! - \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!}$$

$$= n! \sum_{k=0}^{n} (-1)^{k} \frac{1}{k!}$$

$$\longrightarrow n! \cdot (e^{-1}) \sim 0.368 \cdot n!$$

since

$$\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$$

That is, among the $n!$ permutations of n things, about 1/3 have no fixed point.

In fact, *the nearest integer to $n!/e$ is* **exactly** *the number of permutations with no fixed point.*

This is because the exact expression above differs from the infinite series for $n!/e$ by terms whose sum is much less than 1.

That is, (with $f(n)$ the fixed-point-free ones)

$$n! \cdot e^{-1} - f(n)$$

$$(-1)^{n+1} \frac{n!}{(n+1)!} + (-1)^{n+2} \frac{n!}{(n+2)!} + \ldots$$

$$= (-1)^{n+1} \left[\frac{1}{n+1} - \frac{1}{(n+1)(n+2)} + \ldots \right]$$

Estimating that series by a geometric series

$$\frac{1}{n+1} \cdot \sum_{n=1}^{\infty} 4^{-n} = \frac{1}{n+1} \cdot \frac{1/4}{1-1/4} = \frac{1}{3} \cdot \frac{1}{n+1}$$

$$\text{so} \quad \left| \frac{n!}{e} - f(n) \right| << 1$$

The One-Time Pad

If used correctly, the OTP or Vernam cipher is *provably* perfectly secure, and is currently the only known provably secure cipher.

However, it is nearly impossible to use correctly.

If the key is ever *re-used* an OTP degenerates into a **Vigenere** cipher, which is broken (*later*). So **key distribution** is a critical problem.

If the key is not *random* in a *strong-enough sense*, again it degenerates into a sort of Vigenere cipher, and is broken. Making many high-quality random numbers is not so easy.

OTPs *are* used to protect nuclear weapons launch codes and high-level diplomatic traffic, but there key distribution is solved by couriers with sealed diplomatic pouches.

The operation of an OTP is straightforward. To encrypt a message of N characters, we use a *key* of length N, encode characters as integers $0-25$, and (for example)

$$
\begin{aligned}
& i^{\text{th}} \text{ character of ciphertext} \\
& = (i^{\text{th}} \text{ char of plaintext} \\
& \quad + i^{\text{th}} \text{ char of key }) \,\%\, 26
\end{aligned}
$$

Decryption is by the corresponding subtraction and reduction modulo 26. That is, we *add the key to the plaintext like vector addition modulo 26.*

For example, with plaintext

`homefortheholidays`

and key

`pazxqrasdfyipheakl`

the ciphertext is

`WOLBVFRLKJFWAPHAID`

The proof of security is as follows.

The specific claim is that *the* **conditional** *probability that a character of the plaintext is a particular thing* **given** *knowledge of the ciphertext is equal to the probability that that character is that particular thing* (without knowing the ciphertext).

That is, knowing the ciphertext gives us no information about the plaintext.

This *assumes* that the key has never been used before and will not be used again, *and* that the key is *random* in a strong sense.

For example,

$$P(\text{plaintext is } \texttt{horse} | \text{ciphertext } \texttt{XWTHG})$$

$$= \frac{P(\text{plaintxt } \texttt{horse} \ \& \ \text{ciphertxt } \texttt{XWTHG})}{P(\text{ciphertext } \texttt{XWTHG})}$$

$$= \frac{P(\text{plaintxt } \texttt{horse} \ \& \text{ key is } \texttt{XWTHG-horse})}{P(\text{key is } \texttt{XWTHG-horse})}$$

subtracting length 5 vectors modulo 26.

The *randomness* assumption is that any key is equally likely, and certainly is independent of the plaintext, so this is equal to

$$\frac{P(\text{plaintxt } \texttt{horse}) \cdot P(\text{key } \texttt{XWTHG-horse})}{P(\text{key is } \texttt{XWTHG-horse})}$$

$$= P(\text{plaintxt } \texttt{horse})$$

by cancelling.

Again, the formalized version of this says that the *conditional* probability that the plaintext is any particular thing *given* the ciphertext is the same as the probability that the plaintext is that thing.

Randomness

Old or new ciphers are essentially worthless without a good source of random numbers to choose keys, etc.

On linux/unix, `/dev/random` and `/dev/urandom` are processes that attempt to distill good random bytes from processes, keyboard activity, etc.

Even very good pseudorandom number generators (Blum-Blum-Shub, Naor-Reingold) fail in the sense that they can be no better than the random seed and other initial data they use.

Even the very definition of *random* is problemmatical.

Elementary probability does not suffice to define randomness.

For example, the bit string

`1100110011001100110011`

is intuitively *not* random, while maybe

`1111010010000110101001`

is more random.

Yet, if we generate sequences of bits via a fair coin with values `1` and `0` repeatedly (assuming independence) then **every sequence of length 22 is equally likely**, with probability $1/2^{22}$.

That is, the above two strings are equally likely, even though one seems to us to have a *pattern* and the other perhaps does not.

Among many attempts to make rigorous the notion of randomness, the notion of **Kolmogorov complexity** is more successful than most.

Very roughly, in that setting, *a thing is random if it has no shorter description than itself.*

A paraphrase: *a thing is random if it is not* **compressible**.

There is the danger here of subjectivism or relativism, in that the descriptive apparatus and/or the compression apparatus may change.

But a suitably careful formulation of the idea in fact allows proof that a subtler version of this is really well-defined.

For cryptographic purposes, an essentially equivalent intuitive notion is that *the next bit should not be predictable from the previous ones.*

But what does *predictable* mean?

If the sequence is produced by a deterministic process, then it *must* be predictable by the process computing it.

Maybe the idea would be that *lacking a secret* (key) the bits are unpredictable, even if produced by a known deterministic process.

But does it seem possible that zillions of unpredictable bits could be produced from a secret that might consist of just 128 bits?

Shouldn't there be some *conservation of randomness*?