Pólya Counting I

Gordon Royle

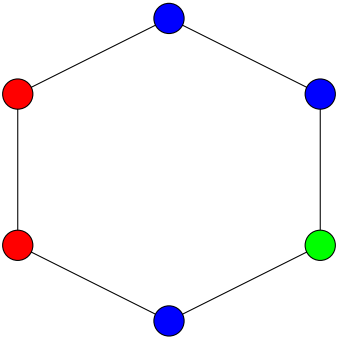
Semester 1, 2004

George Pólya (1887 – 1985)

George Polya discovered a powerful general method for enumerating the number of orbits of a group on particular configurations. This method became known as the Pólya Enumeration Theorem, or PET.

Necklaces

Consider a decorative ornament that consists of n coloured "beads" arranged on a circular loop of string.

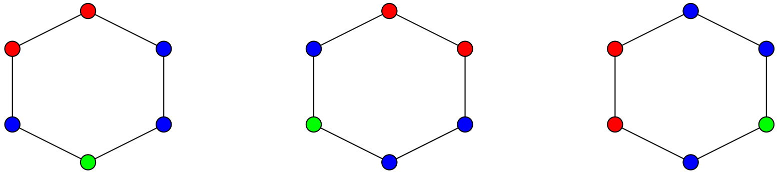


This can be represented simply as a word of length n over a suitable alphabet of colours: $bbgbrr$

Rotation

Two words that differ purely by a (cyclic) rotation clearly represent the same ornament, and are called *equivalent*:

$$rbbgbr \equiv rrbbgb \equiv bbgbrr$$



Necklaces

An $(n, k)-$*necklace* is an equivalence class of words of length n over an alphabet of size k under rotation. The basic enumeration problem is then:

Necklace Enumeration

For a given n and k, how many (n, k)-necklaces are there?

Equivalently, we are asking how many orbits the cyclic group C_n has on the set of all words of length n over an alphabet of size k.

We will denote this value by $a(n, k)$.

$(6,3)$-necklaces

By Burnside's lemma, the number of orbits of C_6 on words of length 6 over an alphabet of size 3 is equal to the average number of words fixed by each element of C_6.

Element g	$\lvert\mathrm{fix}(g)\rvert$
e	3^6
$(1,2,3,4,5,6)$	3
$(1,3,5)(2,4,6)$	3^2
$(1,4)(2,5)(3,6)$	3^3
$(1,5,3)(2,6,4)$	3^2
$(1,6,5,4,3,2)$	3

$$a(6,3) = \frac{1}{6}\sum_{g\in C_6} \lvert\mathrm{fix}(g)\rvert = 130.$$

$(6, k)$-necklaces

If we allow k colours, then the computation is the same except that the number of configurations fixed by each element now depends on k.

Element g	$\lvert\mathrm{fix}(g)\rvert$
e	k^6
$(1,2,3,4,5,6)$	k
$(1,3,5)(2,4,6)$	k^2
$(1,4)(2,5)(3,6)$	k^3
$(1,5,3)(2,6,4)$	k^2
$(1,6,5,4,3,2)$	k

$$a(6,k) = \frac{1}{6} \sum_{g \in C_6} \lvert\mathrm{fix}(g)\rvert = (2k + 2k^2 + k^3 + k^6)/6;$$

(n, k)-necklaces

The number of words fixed by an element g of C_n is completely determined by the number of cycles in the cycle decomposition of g — in fact, if g has c cycles, then it fixes k^c words.
Now, if $g = (1, 2, \ldots, n)$ then the elements of C_n are the n permutations

$$g, g^2, \ldots, g^n = e.$$

The order of the element g^i is

$$n/\gcd(n, i)$$

and hence it has $\gcd(n, i)$ cycles in its cycle decomposition.

(n, k)-necklaces

Therefore the number of (n, k)-necklaces is given by

$$a(n, k) = \frac{1}{n} \sum_{i=1}^{n} k^{\gcd(n,i)}.$$

If p is a *prime number* then this can be substantially simplified, because $\gcd(p, i) = 1$ for all $i < p$, and so we get

$$a(p, k) = \frac{1}{p}((p-1)k + k^p).$$

Euler's ϕ Function

For any positive integer x, let $\phi(x)$ denote the number of integers $1 \leq i \leq x$ such that $\gcd(x, i) = 1$.

This function is sometimes called *Euler's ϕ function* or *Euler's totient function* and its first few values are given below:

x	$\phi(x)$	x	$\phi(x)$
1	1	2	1
3	2	4	2
5	4	6	2
7	6	8	4
9	6	10	4
11	10	12	4

General Result

The following theorem explains the significance of Euler's function for necklaces:

Theorem

For any divisor d of n, there are $\phi(d)$ elements of order d in C_n.

Corollary

The number of (n, k)-necklaces is given by

$$a(n, k) = \frac{1}{n} \sum_{d|n} \phi(d) k^{n/d}.$$

Proof of Theorem

For each possible "greatest common divisor" value $g|n$, how many numbers i are there such that

$$\gcd(n,i) = g?$$

For this to occur, we must have

$$n = n_1 g \qquad i = i_1 g$$

where $\gcd(n_1, i_1) = 1$.

Clearly this number is just $\phi(n_1) = \phi(n/g)$, and putting $d = n/g$ we obtain the result.

In GAP

Using GAP's built-in functions, this can be written in a very slick fashion:

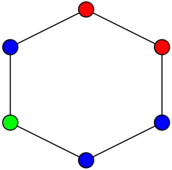
```
neckLaces := function(n,k)
 return Sum(DivisorsInt(n),d->Phi(d)*k^(n/d))/n;
end;
```

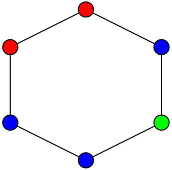
Here `DivisorsInt` returns the list of divisors of a number, and `Phi` is Euler's ϕ function. This use of `Sum` with two arguments applies the function given as the second argument to every element of the list in the first argument and sums the resulting values.

Bracelets

Some of these ornaments can be freely turned over (for example, if the beads are just spherical), but sometimes they cannot, and so whether we consider configurations that are mirror-images to be equivalent or not depends on the application.

An $(n, k)-$*bracelet* is an equivalence class of words of length n under rotation *and* reflection.





Counting Bracelets

In order to determine $b(n, k)$ – the number of bracelets of length n over an alphabet of size k, we need to find the number of orbits of the *dihedral group* D_{2n} on k-ary n-tuples.

Exercise

Determine $b(n, k)$.

Hint: Experiment first with GAP and small dihedral groups. The final expression will differ according to whether n is even or odd, so examine carefully the differences.

Representatives

A necklace was defined to be an equivalence class of words under rotation. For example, if $n = 3$ and the alphabet is $\{0, 1, 2\}$ then the following set is an example of a necklace:

$$\{010012, 100120, 001201, 012010, 120100, 201001\}.$$

Any one of these words suffices to determine the necklace, and so we represent a necklace by using the *lexicographically least* word that it contains. Thus the necklace

$$\{010012, 100120, \textcolor{red}{001201}, 012010, 120100, 201001\}$$

is represented by the word

$$001201.$$

Generation

We could generate all necklaces by generating all the k^n words using the odometer principle and then discarding all the ones that are *not* the lexicographically least in their class.

This is not an efficient way to generate necklaces because it generates k^n words for approximately k^n/n necklaces, thus doing about n times too much work.

An interesting algorithm to generate necklaces was found by Fredericksen, Kessler and Maiorana and is therefore known as the *FKM algorithm*.

FKM Algorithm

To generate all (n,k)-necklaces over the alphabet $\{0,1,\ldots,k-1\}$, consider the following rule to generate a list of words (which will include both necklaces and some non-necklaces) in increasing lexicographic order from the first word $0^n = 000\ldots0$ to the last word $(k-1)^n$.

For any word $\alpha = a_1a_2\ldots a_n$ other than $(k-1)^n$, the successor of α is obtained as follows:

- Let i be the largest value such that $a_i < (k-1)$,
- Let $\beta = a_1a_2\ldots a_{i-1}(a_i+1)$,
- Then $\mathrm{succ}(\alpha)$ is the first n characters of $\beta\beta\beta\ldots$; this word is a necklace if and only if i is a divisor of n.

Example

Suppose we are generating $(6, 3)$-necklaces; then the FKM algorithm starts as follows:

0	0	0	0	0	0	
0	0	0	0	0	1	
0	0	0	0	0	2	
0	0	0	0	1	0	$\leftarrow$ reject
0	0	0	0	1	1	
0	0	0	0	1	2	
0	0	0	0	2	0	$\leftarrow$ reject
0	0	0	0	2	1	
0	0	0	0	2	2	
0	0	0	1	0	0	$\leftarrow$ reject

Example (cont.)

At some later stage of the algorithm, it continues

0	1	2	2	2	2	
0	2	0	2	0	2	
0	2	0	2	1	0	$\longleftarrow$ reject
0	2	0	2	1	1	
0	2	0	2	1	2	
0	2	0	2	2	0	$\longleftarrow$ reject
0	2	0	2	2	1	
0	2	0	2	2	2	
0	2	1	0	2	1	

Pre-necklaces

The set of words produced by the FKM algorithm is actually the collection of *pre-necklaces* — that is, words that can occur at the beginning of a k-ary necklace of some possibly larger length.

Proving this, and the fact that the necklaces occur precisely when i is a divisor of n is not extremely difficult, but does require some careful analysis of the structure of necklaces.

Binary Necklaces

Binary necklaces (i.e those with $k = 2$) can be produced by the FKM algorithm:

0 0 0 0 0 0 0	0 0 0 1 1 0 1	0 0 1 1 1 1 1
0 0 0 0 0 0 1	0 0 0 1 1 1 1	0 1 0 1 0 1 1
0 0 0 0 0 1 1	0 0 1 0 0 1 1	0 1 0 1 1 1 1
0 0 0 0 1 0 1	0 0 1 0 1 0 1	0 1 1 0 1 1 1
0 0 0 0 1 1 1	0 0 1 0 1 1 1	0 1 1 1 1 1 1
0 0 0 1 0 0 1	0 0 1 1 0 1 1	1 1 1 1 1 1 1
0 0 0 1 0 1 1	0 0 1 1 1 0 1	

Unsolved Problem

Is there a list containing representatives of all of the binary n-bit necklaces (for odd n) such that successive elements differ in only one place? (In other words, a Gray code for binary necklaces).

Cycle Index

Given a permutation g of degree n, let $c_i(g)$ be the number of cycles of length i in its cycle decomposition.

Then the *cycle index* of a permutation group is a polynomial that summarizes the information about the cycle types of all the elements of the group.

$$Z_G(X_1, X_2, \ldots, X_n) = \frac{1}{|G|} \sum_{g \in G} X_1^{c_1(g)} X_2^{c_2(g)} \cdots X_n^{c_n(g)}$$

Examples

The cycle index of the group C_6 is given by

$$Z_{C_6}(X_1, X_2, \ldots, X_6) = \frac{1}{6}(X_1^6 + X_2^3 + 2X_3^2 + 2X_6).$$

Notice that the number of $(6, k)$-necklaces is given by

$$a(6, k) = Z_{C_6}(k, k, k, k, k, k)$$

Pólya's Enumeration Theorem

Now we state a simple version of Pólya's Enumeration Theorem.

Pólya's Enumeration Theorem (PET)

Let A and B be two finite sets, and suppose that group G acts on A. Then the number of orbits of G on the set B^A of functions

$$f : A \to B$$

is given by

$$Z_G(|B|, |B|, \ldots, |B|).$$

Necklaces

PET applies directly to necklaces, because if we take

- $A = \{1, 2, \dots n\}$,
- $B = \{0, 1, \dots, k-1\}$, and
- $G = C_n$.

then a function from $A \to B$ corresponds to a word of length n over the alphabet $\{0, 1, \dots, k-1\}$, and the orbits of these words under C_n are precisely the necklaces.

So, we get

$$a(n, k) = Z_{C_n}(k, k, \dots, k)$$

as we have (almost) already seen.

Proof of PET

The proof of PET follows directly from Burnside's lemma.

A function $f : A \to B$ is fixed by a permutation $g \in G$ if and only if f is constant on each of the cycles of g. Therefore if g has $c(g)$ cycles altogether, then there are

$$|B|^{c(g)}$$

functions fixed by g.

Now g makes a contribution to Z_G of

$$X_1^{c_1(g)} X_2^{c_2(g)} \cdots X_n^{c_n(g)}$$

and if we substitute $X_i = |B|$ then we get

$$|B|^{c_1(g)+c_2(g)+\cdots+c_n(g)} = |B|^{c(g)}.$$