From Mathematics to Generic Programming

Brooks Mershon

March 2017

6.5

Solution.

We can refer to the table of nonzero remainders modulo 7 and simply trace the steps taken from repeated applications of a particular element a. The first application is simply a itself, so if we determine that $a \cdot a \cdot a = 1 = e$, then we shall say the order of a is 3.

	1	**2**	**3**	**4**	**5**	**6**	*order*
1	1	2	3	4	5	6	1
2	2	4	6	1	5	5	3
3	3	6	2	5	q	4	6
4	4	1	5	2	6	3	3
5	5	3	1	6	4	2	6
6	6	5	4	3	2	1	2

For example, we have the following progression of repeated applications of the element 5:

$$5 \to 4 \to 6 \to 2 \to 3 \to 1.$$

There are 5 arrows, but we must include 5^1 as the "first" application of 5; the order of the element 5 is 6. Note that the only element with order 1 is the identity element 1.

A larger value of n involves more labor if we choose to continue finding solutions by hand. We might consider a programmatic solution to finding orders of elements in a multiplicative group modulo n. The following example renders a table like the one seen above for $n = 11$. One can see how such a solution lends itself to a generalized solution that obviates the need to use pencil and paper to find the orders of, say, $n = 101$. Of course, even a fairly humble algorithm like the one seen in the implementation below requires a bit of thinking to get right.

https://bl.ocks.org/bmershon/7938f064dc2202364cdd52acbd24805d