

# Exploring Cyclic Relationships

Let's do some fun exploration!

## Cycles, part 1

First, we can do a couple of computations to get a sense of what we might want to study.

- Pick a prime  $p$  and a number  $a$ ,  $1 \leq a \leq p - 1$ .
  - In the console, play around with the powers of  $a$  modulo  $p$ . (The remainder operator in R is `%`.)
  - As you increase the power of  $a$ , do you see any patterns? To what extent do these patterns hold for other choices of  $a$ ?
  - If you don't choose a prime for  $p$ , to what extent do you see the same patterns?

Hopefully you're beginning to see some interesting patterns. We'd like to explore this more, but it's time-consuming to keep typing out repetitive arithmetic.

- Write a function `pow(a, n, p)` that calculates  $a^n$  modulo  $p$ .
  - To improve the runtime of your function, try starting at 1 and repeatedly multiplying an intermediate result by  $a$ , calculating the answer mod  $p$  each time.
  - Using the `timeit` package, quantify the resulting improvement in runtime. How does runtime improve as  $a$  or  $n$  increase in size?
  - Is the runtime improvement merely a constant-factor scaling change (is the new runtime a constant multiple of the previous runtimes)? If not, try to determine the change in [algorithmic time complexity](#). (Skip this for now if you aren't familiar with time complexity or if this is taking too long.)
  - You can improve the runtime further by repeatedly squaring the intermediate result (in a sense, decomposing the final exponent into sums of powers of 2). Implement this and quantify the corresponding runtime improvement.

You may have noticed by now the emergence of a cyclic pattern as you increase  $n$ .

- Now start looking at the *length* of the cycles which emerge. Do you notice any patterns? Pay particular attention to divisibility relations.
- It's still cumbersome to do this with just a single function. Write a function `cyclen(a, p)` that takes as input a prime  $p$  and an integer  $a$  (where

$1 \leq a \leq p - 1$ ) and determines the length of the *cycles* which emerge as the exponent of  $a^n \bmod p$  increases.

- Verify that your function works by reproducing some of the calculations you did earlier.
- Pick a reasonably low prime  $p$  (something under 30 or so) and calculate the corresponding cycle lengths for various values of  $a$  (perhaps all possible values, if your prime is small enough).
- Look for interesting relationships, paying particular attention to (1) what numbers are divisible by all of these cycle lengths and (2) how the aforementioned numbers relate to your prime  $p$ .

Repeat the above analysis for different primes. Talk to other students and the instructors about any results you get.

Putting everything together, try to answer the following questions.

- Suppose that  $p$  is a prime and that we perform all arithmetic modulo  $p$ . We're wondering if any numbers  $a$  correspond to cycles of length  $j$  when we repeatedly exponentiate  $a$  (modulo  $p$ ).
  - Given some  $j$ , when does there exist a cycle of that length for some  $a$ ?
  - Given some  $a$ , what can we say about the relationship between the length of its corresponding exponentiation cycle and  $p$ ?