# 6 Number Theory II: Modular Arithmetic, Cryptography, and Randomness

For hundreds of years, number theory was among the least practical of mathematical disciplines. In contrast to subjects such as arithmetic and geometry, which proved useful in everyday problems in commerce and architecture, astronomy, mechanics, and countless other areas, number theory studies very abstract ideas called numbers, and applications of the subject are not immediate. Over the course of the second half of the twentieth century, however, number theory became increasingly more applicable, and today make possible a wide range of technologies. In this section we will consider **modular arithmetic** and applications to **cryptography** and to generating "**random numbers**" by deterministic computers.

## 6.1 Introduction to Cryptography

Since ancient times, people desiring to transmit messages privately have devised methods of encoding messages, so that no person but the intended recipient could read the message. The ability to successfully encode and decode messages has played a central role in the development of financial markets and in history-altering military turnarounds. We use **cryptography** to refer to the study of how information can be made secretive enough so that bad people can't read it, yet still accessible enough so that good guys can. Cryptography is a very exciting and developing area of contemporary mathematics, with connections to number theory and computational complexity.

Let us consider a person Alice who would like to send a secret message to another person Bob. Perhaps Alice and Bob are childhood friends and are planning a surprise birthday party for a mutual friend. Or perhaps Alice and Bob have never met, but Alice would would like to send Bob her credit card information so she can pay for something Bob is selling. In both cases, Alice and Bob would like to guarantee several things: (a) Alice would like to ascertain that Bob has received her message; (b) both Alice and Bob would like to know that no one else has seen the secret message; (c) Bob would like to ascertain that the message he believes to have come from Alice has indeed come from Alice. It is not immediately clear how we can guarantee each of these except in the case where Alice and Bob actually meet up and Alice whispers the message into Bob's ear. What should they do, however, if they are far apart? If they send a message through the postal service, there is a small chance that (notwithstanding the serious federal crime involved in opening someone else's mail) that an eavesdropper might intercept the message before it reaches Bob. Even if they use the telephone, or an email, or a text, there is a chance that the intended message and information will make its way to the wrong hands. These kinds of questions motivate the need to develop methods of encoding and decoding information so that messages can be communicated securely.

We briefly note several methods used to solve some of the above problems. Bob can send back a note saying "I received your message", though the same security concerns relevant to the initial message will be relevant here as well. Signing one's signature to a piece of a paper is a relatively simple way in which Alice can convince that the message indeed came from her. This is partly because while reading and identifying a signature is relatively easy, actually creating it is complicated, for all except the person signing it (though of course signatures can be forged). In this section, we will focus mostly on the problem (b), that is, how can we ensure that no eavesdropper can read the message intended solely for Bob.

Simple ways of encoding messages were known since antiquity. Sometimes letters were switched for other letters, or for numbers, and so an eavesdropper quickly looking at an encoded message would only see gibberish. However, this approach has many limitation. For starters, how would Alice communicate to Bob the scheme which she used to encode the message and which he, consequently, will need to decode it? If he can determine this by himself, perhaps through some guesswork, then what would stop someone else from doing the same? Many somewhat sophisticated methods have been developed over the centuries for encoding and decoding secret messages, though in this section we will focus on one that is built on what is called **modular arithmetic**, a system of arithmetic that in some sense only has a finite number of numbers.

## 6.2 Modular Arithmetic

Every reader is familiar with arithmetic from the time they are three or four years old. It is the study of numbers and various ways in which we can combine them, such as through addition and subtraction, multiplication and division. Since even before they were in grade school, every reader knew that adding 2 and 2 together gives us 4, and can make that calculation now without almost any thinking. And even if the answer is not immediately obvious, every college student (at least in Penn), knows how to add together much larger numbers, such as 4,378,123 and 5,621,877. This is classical arithmetic, and it turns up in countless applications in our everyday lives.

The reader is also likely familiar with another kind of arithmetic, even if we don't always think of it as such. If it is 4 o'clock now, what will the time be in 25 hours? If we didn't know from watches and clocks, we would probably have answered 29 o'clock. But we are familiar with watches, clocks, and the standard conventions of time-keeping, and so every reader would probably have answered the answer with 5 o'clock. How can we add 25 to 4 and end up with 5? The reason is that in this system 25 o'clock is the same as 1 o'clock, 26 is the same as 2, and so forth. In many time-keeping systems, we don't even use numbers larger than 12, and instead use a.m. and p.m. (from the Latin *ante meridiem* and *post meridiem*) to denote the earlier and latter halves of a 24-hour period. Such systems, that "wrap around" after hitting some limit, are called **modular arithmetic systems**, and play an important role both in theoretical and applied mathematics.

Modular arithmetic motivates many questions that don't arise when studying classic arithmetic. For example, in classic arithmetic, adding a positive number $a$ to another number $b$ always produces a number larger than $b$. In modular arithmetic this is not always so. For example, if it is now 4 o'clock and we "add" 23 hours, the time will then be 3 o'clock, which doesn't appear to be larger than 4 o'clock. In fact, it is no longer clear whether it makes sense at all to discuss "larger" and "smaller" in such systems.

Here is another question. Suppose it is now 2 o'clock and we wait for 1 hour and then write down the time. We then wait another hour and mark the time, and repeat this until we eventually mark 2 o'clock again, at which point we stop. It is clear that when we stop, we will have marked down every hour. If we do the same thing but instead wait 2 or 3 hours in between each marking there will be certain hours which we never mark, such as 7 o'clock. But if we wait 5 hours between each marking, then we will eventually mark every hour. This raises the question, for which waiting intervals between marks can we ensure that we will eventually mark every hour?

While this particular example may seem contrived, it should motivate us think, if even momentarily, about modular arithmetic systems and the ways in which they are similar to and different from the classical arithmetic with which we are familiar. The next several sections will investigate these systems which have a finite number of numbers, and in which numbers "wrap around" after going too high.

The central definition in studying modular arithmetic systems establishes a relationship between pairs of numbers with respect to a special number $m$ called the *modulus*:

**Definition 25.** *Two integers $a$ and $b$ are* **congruent modulo** *$m$ if they differ by an integer multiple of $m$, i.e., $b - a = km$ for some $k \in \mathbb{Z}$. This equivalence is written $a \equiv b \pmod{m}$.*

Although this definition looks somewhat technical, the idea is very simple. For some fixed integer $m$, two numbers are roughly the same if they differ by multiples of $m$. In a sense, this definition generalizes previous discussions of odd and even numbers. In previous sections, we proved theorems such as the square of an even number is even and the square of odd number is odd. As far as even and odds numbers go, and as far as these theorems are concerned, there is no difference between 17 and 2073, as both are odd and behave the same under squaring. In a similar manner, in modular arithmetic, there is no difference between a pair of numbers that differ by the modulus $m$, which could be 2 or could be 15,485,863. In arithmetic mod 7, for example, there is no difference between 1, 8, and 15, as they all differ from one another by multiples of 7. Likewise, 22, 701 and -6 also differ from all of these numbers by multiples of 7, and are hence congruent.

**Example 1.** Every number is congruent to itself for any modulus; that is, $a \equiv a \pmod{m}$ for any $a, m \in \mathbb{Z}$. The reason for this is that $a - a = 0$, which is a multiple of $m$, since $0 = 0 \times m$ for any $m$. It might seem a bit silly, but is a consequence of the way in which we defined congruence.

**Example 2.** Every number is congruent to any other number mod 1; that is, $a \equiv b \pmod{1}$ for any $a, b \in \mathbb{Z}$. The reason for this is that $b - a$, is a multiple of 1 for any $a$ and $b$. Again, this might seem a bit silly, but is a consequence of the way in which we defined congruence.

**Example 3.** Any even numbers are congruent to one another mod 2; likewise, any odd numbers are congruent to one another mod 2. For example, we have $12 \equiv 3132 \pmod{2}$ and $-7 \equiv 19 \pmod{3}$. This is because any pair of even numbers differ from one another by a multiple of 2. Likewise, any pair of odd numbers differ from one another by a multiple of 2.

**Example 4.** The numbers 31 and 46 are congruent mod 3 because they differ by a multiple of 3. We can write this as $31 \equiv 46 \pmod{3}$. Since the difference between 31 and 46 is 15, then these numbers also differ by a multiple of 5; i.e., $31 \equiv 46 \pmod{5}$.

**Example 5.** By the definition of congruence, every pair of integers $a$ and $b$ are congruent mod 1, since any pair of integers differ by a multiple of 1. In symbols, for all integers $a$ and $b$, we have $a \equiv b \pmod{1}$.

**Example 6.** In general it is not true that $a \equiv -a \pmod{m}$, unless $m = 2$ or else $a$ is a multiple of 2. For example, it is not true that $7 \equiv -7 \pmod{3}$, since the difference between 7 and -7 is 14, which is not a multiple of 3.

**Rules of Modular Arithmetic**

After considering the basic definition of modular arithmetic, we next consider some of its basic properties. It turns out that modular arithmetic follows many of the same rules of classical arithmetic, thus making it very easy to work with. In order to highlight what is going on, we try to compare and contrast modular arithmetic to classical arithmetic.

Suppose we have two numbers $a$ and $b$:

$$\begin{aligned} a &= 5 \\ b &= 8. \end{aligned}$$

We all know that in classical arithmetic we can combine these equations to obtain:

$$a + b = 5 + 8 = 13.$$

More generally, if we have

$$\begin{aligned} a &= c \\ b &= d, \end{aligned}$$

then we can combine them in many different ways, to obtain:

$$\begin{aligned} a + b &= c + d, \\ a - b &= c - d, \\ a \times b &= c \times d. \end{aligned}$$

Pause to think about this statement, and make sure it aligns with what you know. Of course these are only several ways of combining these equations, and every reader can think of several others. All of the above are "rules" of classical arithmetic. What we would like to do now is consider whether similar rules apply to modular arithmetic as well.

Suppose we have the following two congruence relations:

$$\begin{aligned} a &\equiv b \pmod{m} \\ c &\equiv d \pmod{m}. \end{aligned}$$

Are we able to combine these to obtain

$$\begin{aligned} a + b &\equiv c + d \pmod{m}, \\ a - b &\equiv c - d \pmod{m}, \\ a \times b &\equiv c \times d \pmod{m}? \end{aligned}$$

That is, do the rules that govern how we can combine equations in classical arithmetic also govern the ways in which we combine statements in modular arithmetic? In what follows we prove that indeed many of the rules *do* carry over – the rules of modular arithmetic will be familiar to us.

## Addition

The first rule we consider is that associated with addition. Suppose we have two congruence relations: $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. In other words, $a$ and $b$ are congruent and $c$ and $d$ are congruent, both mod $m$. We can add the left sides of these congruent relations, add the right sides, and the results will again be congruent. In symbols,

**Theorem 15.**

$$\begin{array}{lrcll} \textit{If} & a & \equiv & b \pmod{m} & \textit{and} \\ & c & \equiv & d \pmod{m}, & \textit{then} \\ & a+c & \equiv & b+d \pmod{m}. & \end{array}$$

Proving this result involves nothing more than applying the definition of congruence and some basic algebraic manipulation.

*Proof.* By the definition of congruence (Definition 25) we know that $a$ and $b$ differ by some multiple of $m$, i.e.,

$$b - a = km \tag{64}$$

for some $k \in \mathbb{Z}$. Likewise we know that $c$ and $d$ also differ by some multiple of $m$, i.e.,

$$d - c = jm \tag{65}$$

for some $j \in \mathbb{Z}$. Note that we use $j$ instead of $k$ since the multiple of $m$ by which $c$ and $d$ differ might be different from the multiple by which $a$ and $b$ differ. Next we add these two equations together:

$$(b - a) + (d - c) = km + jm. \tag{66}$$

We can rewrite this equation as

$$(b + d) - (a + c) = (j + k)m. \tag{67}$$

By the definition of congruence modulo $m$, this is the same as saying that $a + c$ is congruent to $b+d$ modulo $m$, since $a+c$ and $b+d$ differ by an integer multiple $(j + k)$ of $m$. In symbols, we have:

$$a + c \equiv b + d \pmod{m}, \tag{68}$$

as desired. □

A similar proof can be used to show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a - c \equiv b - d \pmod{m}$.

These two results allow us to treat all numbers that are congruent modulo $m$ as identical when adding and subtracting numbers. If we know that $a \equiv 3 \pmod{7}$ and $b \equiv 4 \pmod{7}$, then we can know that $a + b \equiv 7 \equiv 0 \pmod{7}$. This is true whether $a$ is 10 or 703, and whether $b$ is 7004, 10000, or 7,000,004. What $a$ and $b$ actually are does not matter if we only want to determine whether $a + b$ is congruent to 0 or not.

### Multiplication

After understanding how addition and subtraction work in modular arithmetic, we turn our attention to understanding multiplication. In classical arithmetic, if $a = 2$ and $b = 5$, then of course $a \times b = 2 \times 5 = 10$. Does a similar relationship also hold in modular arithmetic? In particular, if we know that $a \equiv 2 \pmod{m}$ and $b \equiv 5 \pmod{m}$, do we know that $a \times b \equiv 2 \times 5 \pmod{m}$?

The following theorem answers this question affirmatively.

**Theorem 16.**

$$\begin{array}{lrcll} \textit{If} & a & \equiv & b \pmod{m} & \textit{and} \\ & c & \equiv & d \pmod{m}, & \textit{then} \\ & a \times c & \equiv & b \times d \pmod{m}. & \end{array}$$

*Proof.* By the definition of congruence we know that $a$ and $b$ differ by a multiple of $m$, as do $c$ and $d$:

$$\begin{aligned} b - a &= jm \\ d - c &= km \end{aligned}$$

for some $j, k \in \mathbb{Z}$. Note that we use distinct multiples $j$ and $k$ for the two equations, since $a$ and $b$ might differ by one multiple of $m$, and $c$ and $d$ might differ by another multiple of $m$.

To prove the desired result, we rearrange the equations:

$$\begin{aligned} b &= jm + a \\ d &= km + c \end{aligned}$$

We multiply both sides by each other to obtain

$$\begin{aligned} bd &= (jm + a)(km + c) \\ &= jkm^2 + jmc + kma + ac \\ &= (jkm + jc + ka)m + ac. \end{aligned}$$

We then subtract $ac$ from both sides to obtain

$$bd - ac = (jkm + jc + ka)m.$$

Since $(jkm + jc + ka)m$ is an integer multiple of $m$, then $ac$ and $bd$ differ by an integer multiple of $m$, and so by definition are congruent mod $m$. □

**Example 1.** If we know that $a \equiv 3 \pmod{7}$ and we know that $b \equiv 4 \pmod{7}$, then we can determine that $ab \equiv 12 \equiv 5 \pmod{7}$. This is true whether $a$ is 10, 703, or 7,000,003 and whether $b$ is 7004 or 10000. In any of these cases, the product $ab$ will be congruent to 5 modulo 7.

**Example 2.** How can we simplify $20 \times 21$ in arithmetic modulo 19? We first note that $20 \equiv 1 \pmod{19}$ and also that $21 \equiv 2 \pmod{19}$. Theorem 16 tells us that we can combine these equations to obtain $20 \times 21 \equiv 1 \times 2 \equiv 2 \pmod{19}$.

**Example 3.** Can we simplify $17^{753}$ in arithmetic modulo 9? We first note that $17 \equiv -1 \pmod 9$, because 17 and -1 differ by a multiple of 9. Theorem 16 allows us to then combine this congruence relation as many times as we would like. In particular, by combining 753 copies, we obtain $17^{753} \equiv (-1)^{753} \pmod 9$. Since $(-1)^n = -1$ for any odd integer $n$, we have $17^{753} \equiv -1 \pmod 9$. Finally, if we would like to have a simple, positive answer, then we can add 9 to obtain a final answer of 8.

Theorems 15 and 16 show us that we can treat all numbers that are congruent modulo $m$ as the same, in addition and in multiplication operations. Division is much more complicated, and will not be discussed.

## Remainders

We take a moment to draw out a connection to division with remainders, an idea we considered briefly in Section 4.1. In particular, back in elementary school we learned about a way of dividing integers by other integers that entirely avoids decimals and fractions. In particular, suppose we divide 7 by 4. In third, fourth, or fifth grade, we learned that we can write this as 1, remainder 3. That is, 4 can 1 time “into” 7, leaving over 3. As we got older, we learned that we could also write the answer as 1.75 or 1¾, but we still occasionally deal with situations in which discussing fractions would be silly. If we have 52 playing cards and 5 players, a dealer could give each player 10 cards and then be left with 2 cards. It makes little sense to say that the dealer should give each player 10.2, or 10 and a fifth, cards.

What is the connection of modular arithmetic to division with remainders? Suppose that we divide some integer $a$ by another integer $m$. Notice that the “remainder” is always congruent to $a$ modulo $m$. For example, suppose we divide 1031 by 19. We obtain 54, remainder 5. This tells us that 5 is congruent to 1031 modulo 19. Likewise, since the remainder of 7381/57 is 28, we know that $28 \equiv 7381 \pmod{57}$.

Why is the remainder after division always congruent to the number we are dividing? One way to think about this is by considering how we can find a remainder without actually doing any division. Suppose we want to know the remainder of 11 after dividing by 3. We can subtract 3 over and over until we obtain a number that is smaller than 3: 11, 8, 5, and eventually 2. Each time we subtract 3, we are realizing that 3 can “go into” 11 one more time; whatever is left at the end is the remainder. At the same time, we got from the original number to the remainder by jumps of 3, so of course the difference between 11 and 2 is divisible by 3, making 11 and 2 congruent. The same idea works for dividing any number $a$ with any other number $m$.

**Standard Representation**

We have by now seen that in arithmetic modulo $m$, there is no difference between writing 1, $1+m$, $1+2m$, and so forth, at least as far as addition, subtraction, and multiplication are concerned. For this reason, writing $4+11 \equiv 15 \pmod{13}$ is "just as correct" as writing $4+11 \equiv 2 \pmod{13}$, and "just as correct" as writing $4+11 \equiv -11 \pmod{13}$. As far as arithmetic modulo 13 is concerned, 2, 15, and -11 are exactly the same number. However, in some applications it is convenient to agree upon a standard way to represent numbers. What is a good way to do this? Which of $\{\ldots, a-2m, a-m, a, a+m, a+2m, \ldots\}$ should we consider the standard representative?

You have likely encountered a similar problem back in your days learning about trigonometric functions. A teacher may have asked you what is the inverse sine of $-1$, i.e., $\sin^{-1}(-1)$. You may have correctly answered $270°$. Or you may have correctly answered $-90°$. In fact, any number that can be written $270° + n360°$, for any integer $n \in \mathbb{Z}$, would also be equally correct. But if each student wrote a different number on an exam, it could take a long time to determine whether or not every answer is correct. Is $1500°$ a correct solution? Is $1530°$? For this reason, we might specify that we looking for a correct answer between $0°$ and $360°$, or else between $-180°$ and $180°$, since there is exactly one correct answer in each of these ranges.

In the same way, when working in arithmetic modulo 41, the numbers $\{\ldots, -29, 12, 53, 94, 135, \ldots\}$ are all the same, yet we might hope to specify one of them to be the standard representation of them. Indeed, in arithmetic modulo $m$, we refer to the numbers $\{0, 1, 2, \ldots, m-1\}$ as the **standard representations** of the integers. If numbers are always represented in this standard form, determining whether or not two numbers are congruent is as easy as looking at whether the numbers are equal. Notice also that this set of numbers is also the set of possible remainders after dividing a number by $m$.

**Example 1.** Suppose we want to know the remainder of $17 \times 18$ when it is divided by 19. We can do this in two different ways. First, we can multiply the two numbers directly and obtain 306; some calculation will show that 306 is congruent to 2 modulo 19. Alternatively, we know that $17 \equiv -2 \pmod{19}$ and $18 \equiv -1 \pmod{19}$. Multiplying both sides we see that $17 \times 18 \equiv (-2) \times (-1) \equiv 2 \pmod{19}$.

**Example 2.** Suppose we want to determine the standard form of $17^2$ in mod 19 arithmetic. One way in which we can do this is by considering the square of 17, which is 289, divide that by 19 and then take the remainder. However, since we know that $17 \equiv -2 \pmod{19}$, we can multiply this congruence equation by itself to obtain $17^2 \equiv -2^2 \equiv 4 \pmod{19}$. We can easily verify that the remainder of 289, when divided by 17, is indeed 4.

**Example 3.** Suppose we want to determine the standard form of $18^{489391312}$ in mod 19 arithmetic. We should first notice that in mod 19 arithmetic, 18 is congruent to $-1$, and so $18^{489391312} \equiv (-1)^{489391312} \pmod{19}$. It is relatively easy to see that if $n$ is odd then $(-1)^n = -1$, and if $n$ is even then $(-1)^n = 1$. Since 489391312 is even, $18^{489391312} \equiv 1 \pmod{19}$.

## Dividing by 9

We can use the rules of modular addition and multiplication to prove a theorem you may have once seen. Suppose we have a number, for example 2,383,623, and want to know whether it is divisible by 9. Is there an easy way to figure this out without doing "long division"? You may have learned the following trick: add up the digits of the number (e..g., $2 + 3 + 8 + 3 + 6 + 2 + 3 = 27$). If this sum is divisible by 9, then so is the original number; if the sum is not divisible by 9, then neither is the original number. Is this just a miraculous trick, or is it something that we can prove should work?

The rules of modular addition and multiplication (Theorems 15 and 16 above) can help us prove this beautiful result. Let's begin by proving a simpler result about the remainders we get when we divide powers of 10 by 9. In particular, the remainder is always 1.

**Lemma 17.** *For any natural number $n$, we have $10^n \equiv 1 \pmod{9}$.*

*Proof.* Recall that if we have two congruences: $a \equiv b$ and $c \equiv d \pmod{m}$, then we can combine them to form a new congruence relation: $ac \equiv bd \pmod{m}$. Since $10 \equiv 1 \pmod{9}$, then we can combine the equation with itself to obtain $100 = 10 \times 10 \equiv 1 \times 1 \equiv 1 \pmod{9}$. We can indeed combine this equation with itself as many times as we want (e.g., $n$ times), and therefore have $10^n \equiv 1^n \equiv 1 \pmod{9}$ for any natural number $n$. □

Next, let's consider what happens when we divide numbers such as 300, 5000, and 2,000,000 by 9. What are the remainders? Theorem 16 can help us see that the remainders are 3, 5, and 2 in these examples. To see why this is so, notice that each of these numbers can be written as the product of an integer and a power of 10: $300 = 3 \cdot 10^2$, $5000 = 5 \cdot 10^3$, and 2,000,000$= 2 \cdot 10^6$. This leads us to the following theorem.

**Lemma 18.** *For any natural numbers $c$ and $n$, we have $c \cdot 10^n \equiv c \pmod{9}$.*

*Proof.* Recall that if we have two congruences: $a \equiv b$ and $c \equiv d \pmod{m}$, then we can combine them to form a new congruence relation: $ac \equiv bd \pmod{m}$. Since $c \equiv c$ and $10^n \equiv 1 \pmod{9}$ for any $n$, then we can combine the equations to obtain $c \cdot 10^n \equiv c \cdot 1 \equiv c \pmod{9}$. □

This now leads us to our central theorem:

**Theorem 19.** *A number is divisible by 9 if and only if the sum of its digits (written in base 10) is divisible by 9.*

*Proof.* In base 10, every number can be written as a sum of ones, tens, hundreds, thousands, and so forth. For example, 5776 = 5000+700+70+6. More generally, we can write this as $n = c_0 + c_1 10^1 + c_2 10^2 + c_3 10^3 + \ldots$, where the $c_i$ variables are the numbers of ones, tens, hundreds, thousands, and so forth. According to Lemma 18, for each of the $c_i$ we have $c_i \cdot 10^n \equiv c_i \pmod 9$. Using Theorem 15, we can combine the congruence relations

$$\begin{array}{rcll} c_0 & \equiv & c_0 & \pmod 9, \\ c_1 & \equiv & c_1 10^1 & \pmod 9, \\ c_2 & \equiv & c_2 10^2 & \pmod 9, \\ c_3 & \equiv & c_2 10^3 & \pmod 9, \\ & \dots & & \\ c_n & \equiv & c_2 10^n & \pmod 9, \end{array}$$

to give us

$$c_0 + c_1 10^1 + c_2 10^2 + \dots c_n 10^n \equiv c_0 + c_1 + c_2 + \dots c_n \pmod 9 \tag{69}$$

In other words, a number $n$ is congruent to the sum of its digits in mod 9. If a number is divisible by 9, i.e., $n \equiv 0 \pmod 9$, then so is the sum of its digits. $\square$

## 6.3 Modular Exponentiation

Most technological applications of modular arithmetic involve exponentials with very large numbers. For example, a typical problem related to encryption might involve solving one of the following two equations:

$$\begin{aligned} 67930^{32319} &\equiv a \pmod{103969} \qquad (70)\\ 67930^{b} &\equiv 48560 \pmod{103969}. \qquad (71)\end{aligned}$$

It turns out that $a = 6582$ and $b = 32320$ solve these equations, but those answers are not obvious at all from looking at the equations. More importantly, it is not even clear *how* we would go about determining $a$ and $b$. In what is part of a great mystery of the modern study of computational complexity, the first equation is relatively easy for computers to solve, whereas there is no known way of efficiently solving the second problem. In this section we will look at some problems involving modular exponentiation and some techniques we can use to solve such problems.

Suppose we are asked to determine the remainder of the enormous number $10^{51239203}$ after dividing it by 5. This number has over 50 million digits! How on earth can we hope to ever figure out such a difficult problem without a calculator that can hold more than 8 or even a few dozen digits? Although this might appear impossible to solve, you might notice that 10 is divisible by 5, and the enormous number is just a multiple of 10. If the remainder of 10 when divided by 5 is 0, then so is any multiple of 10, including the enormous number. Of course the answer would be the same if we were attempting to divide it by 2 instead, but what would happen if we divide it by 3, 7, or some other number?

### Patterns

We begin by considering how to search for patterns among the remainders when we taken a number to subsequently higher powers. For example, let us consider the remainders of 10, 100, 1000, and so forth when we divide them by 3. The first thing we notice is that the remainder of 10 after dividing it by 3 is 1. In the language of modular arithmetic we can write:

$$10^1 \equiv 1 \pmod{3}. \qquad (72)$$

The exponent next to the 10 is not necessary but we place it there to make the next step slightly easier. Say that at this point we want to determine the remainder of 100 after dividing it by 3. There are two ways we can go about doing this. First, we can do simple arithmetic to determine that 100/3 equals 33, remainder 1. Although this calculation is not terribly difficult, we can actually avoid it using a rule we saw in the previous section. Namely, if we have two congruence relations, then we can combine them by multiplying both left-hand sides and both right-hand sides to obtain a new congruence relation:

**Theorem.**

$$\begin{array}{lrcll} \textit{If} & a & \equiv & b \pmod{m} & \textit{and} \\ & c & \equiv & d \pmod{m}, & \textit{then} \\ & a \times c & \equiv & b \times d \pmod{m}. & \end{array}$$

In our particular case, we know that

$$\begin{array}{rcll} 10^1 & \equiv & 1 \pmod{3}, & \text{and} \\ 10^1 & \equiv & 1 \pmod{3}. & \end{array}$$

Of course these are the same equation, but writing them out in this way allows us to think of them in terms of the previous theorem. More specifically, this theorem allows us to multiply both sides of the equation together, to get:

$$\begin{array}{rcll} 10^1 \times 10^1 & \equiv & 1 \times 1 & \pmod{3}, \\ 10^2 & \equiv & 1 & \pmod{3}. \end{array}$$

We can then use the same technique, through induction, to show that *all* integer powers of 10 are congruent to 1 mod 3, since we can continue multiplying our resulting equation by the initial equation $10^1 \equiv 1 \pmod{3}$. In other words, all positive integer powers of 10, when divided by 3, give us a remainder of 1!

We have chosen a relatively simple case to highlight the usefulness of Theorem 2 for simplifying what might otherwise be very complicated calculations. We now consider several more complex examples in which we can determine patterns as we consider $a^n \pmod{m}$ as $n$ increases.

**Example 1.** Consider the very large number $7^{1383921}$ and how we might determine its remainder after dividing it by 4. Of course we know that the only possible remainder are 0, 1, 2, and 3, but it is not clear how to determine which of those it is. Simple calculations show the following pattern:

$$\begin{array}{rcll} 7^1 & \equiv & 3 & \pmod{4}, \\ 7^2 & \equiv & 1 & \pmod{4}, \\ 7^3 & \equiv & 3 & \pmod{4}, \\ 7^4 & \equiv & 1 & \pmod{4}, \ldots \end{array}$$

It seems that if $n$ is odd, then $7^n \equiv 3 \pmod{4}$, and if $n$ is even, then $7^n \equiv 1 \pmod{4}$. We can prove that this pattern will repeat as $n$ increases by noticing that $7^2 \equiv 1 \pmod{4}$. Combining this with Theorem 16 shows that if $7^n \equiv 3 \pmod{4}$ then $7^{n+2} \equiv 3 \pmod{4}$, and likewise if $7^n \equiv 1 \pmod{4}$ then $7^{n+2} \equiv 1 \pmod{4}$. Therefore, the pattern repeats with a period of 2. Determining the remainder of $7^{1383921}$ when dividing by 4 is then straightforward – since the exponent $n = 1383921$ is odd, the remainder must be 3.

**Example 2.** Let us consider the very large number $4^{2349321230}$ and determine its remainder after dividing it by 15. Of course we know that the only possible solutions are in $\{0, 1, 2, \ldots, 14\}$, but that is still a wide range of options, and it is not clear how to determine which of those it is. Simple calculations show the following pattern:

$$\begin{aligned} 4^1 &\equiv 4 \pmod{15}, \\ 4^2 &\equiv 1 \pmod{15}, \\ 4^3 &\equiv 4 \pmod{15}, \\ 4^4 &\equiv 1 \pmod{15}, \ldots \end{aligned}$$

It seems that if the exponent $n$ is odd, then $4^n \equiv 4 \pmod{15}$, and if $n$ is even, then $4^n \equiv 1 \pmod{15}$. This pattern too will repeat ad infinitum, because in this case we have $4^2 \equiv 1 \pmod{15}$, and so increasing the exponent $n$ by 2 will never change the remainder mod 15, and $4^n \equiv 4^{n+2} \pmod{15}$ for all exponents $n$. Determining the remainder of $4^{2349321230}$ when dividing by 15 is then straightforward – since the exponent $n = 2349321230$ is even, the remainder must be 1.

**Example 3.** The particular patterns need not have a length of 2, and indeed most of the time they don't. Here we consider a repeating pattern with a slightly longer period. Let us consider the very large number $7^{30001}$ and determine its remainder after dividing by 18. Simple calculations show the following pattern:

$$\begin{aligned} 7^1 &\equiv 7 \pmod{18}, \\ 7^2 &\equiv 13 \pmod{18}, \\ 7^3 &\equiv 1 \pmod{18}, \\ 7^4 &\equiv 7 \pmod{18}, \\ 7^5 &\equiv 13 \pmod{18}, \\ 7^6 &\equiv 1 \pmod{18}, \ldots \end{aligned}$$

Here the pattern repeats every 3, because $4^3 \equiv 1 \pmod{18}$ and so increasing $n$ by 3 will never change the remainder mod 18. Determining the remainder of $7^{30001}$ when dividing by 18 then requires us to look at the exponent $n = 30001$. Since adding and subtracting multiple of 3 from this number will not change the remainder, we should subtract from it 30000, which of course is a multiple of 3. We can then determine that $7^{30001} \equiv 7^1 \equiv 7 \pmod{18}$.

**Example 4.** Here we consider a repeating pattern with a period of 4. Let us consider remainders of all numbers $5^n$ after dividing them by 13. Simple calculations show the following pattern:

$$\begin{aligned} 5^1 &\equiv 5 \pmod{13}, \\ 5^2 &\equiv 12 \pmod{13}, \\ 5^3 &\equiv 8 \pmod{13}, \\ 5^4 &\equiv 1 \pmod{13}, \\ 5^5 &\equiv 5 \pmod{13}, \\ 5^6 &\equiv 12 \pmod{13}, \ldots \end{aligned}$$

Here the pattern repeats every 4 powers, since $5^4 \equiv 1 \pmod{13}$. Therefore, increasing the exponent $n$ by 4 will never change the remainder when dividing by 13, and $5^n \equiv 5^{n+4} \pmod{13}$ for all exponents $n$. Determining the remainder of $5^n$ when dividing by 13 then requires us to determine whether the exponent $n$ is divisible by 4. If it is divisible by 4, then the remainder must be 1. Otherwise, if the remainder is 1, then $5^n \equiv 5 \pmod{13}$; if the remainder is 2, then $5^n \equiv 12 \pmod{13}$; and if the remainder is 3, then $5^n \equiv 8 \pmod{13}$.

### Maximum Length of Patterns

Every sequence of powers $a^1, a^2, a^3, \ldots \pmod{m}$ eventually forms a repeating pattern, though the length of these patterns can be significantly larger than 4. Here we consider the question – how long can the period of such a pattern be? So far we have seen patterns of periods 1, 2, 3, and 4. In all cases, the length of the period was smaller than the modulus $m$. Was this coincidental? Can a repeating pattern have a period longer than the modulus?

To see that the maximum length of a repeating pattern is $m-1$, we first point out that there are only $m$ possible remainders when dividing by $m$: $0, 1, 2, \ldots m-1$. Second, we note that if 0 appears anywhere in the pattern, then all subsequent remainders must be 0. To understand why this is true, consider a number $a$ and some power $n$ for which

$$a^n \equiv 0 \pmod{m}. \tag{73}$$

The next number in the pattern is the remainder of $a^{n+1}$ after dividing it by $m$. Of course it is always true that

$$a \equiv a \pmod{m}, \tag{74}$$

since a number is always congruent to itself. Theorem 16, which we have already seen several times, allows us to combine these two equations to obtain:

$$a^n \times a \equiv 0 \times a \pmod{m},$$

and so

$$a^{n+1} \equiv 0 \pmod{m}.$$

The same technique can be used to show that $a^{n+2}, a^{n+3}, \ldots$ are all congruent to 0 mod $m$, and so all subsequent powers must be congruent to 0.

Therefore, a repeating pattern that does not consist merely of 0's can only contain the $m-1$ distinct numbers: $1, 2, \ldots m-1$. Next, it is easy to see that any of these $m-1$ numbers can appear at most once in a repeating pattern. It is not possible, for example, to have a repeating pattern 2, 3, 2, 1 that repeats itself over and over. Why not? Each consecutive term in the sequence can be calculated from the term before it, by multiplying it by $a$. If we multiply 2 by $a$, the result can either be 3 or it can be 1, but it can't be both. So if 2 is followed by 3 in the pattern, then it must always be followed by 3, and it cannot sometimes be followed by a 1. Since each number is always followed by the same number, once we return to a number we have seen before, the pattern will begin to repeat again. The longest possible pattern then includes all integers between 1 and $m-1$, but not 0, as explained. Therefore, if we are dividing powers of $a$ by $m$, then the maximum length of a repeating pattern of remainders is $m-1$.

To see that this is indeed possible, consider the remainders of $5^1, 5^2, 5^3, \ldots$ when divided by $m = 277$. We obtain: 5, 25, 125, 71, 78, 113, 11, 55, …; the pattern will not repeat before we reach $5^{277}$, which is congruent to 5 and which thus begins the pattern again. Now that we are aware of patterns with very long periods, the approach of finding short patterns will not always help us simplify large exponents. Fermat's Little Theorem gives us an alternate shortcut for computing modular remainders of large exponents.

## Fermat's Little Theorem

As we have seen, every sequence of powers $a^1, a^2, a^3, \ldots \pmod{m}$ will eventually form a repeating pattern, which can be as long as $m-1$. If the length of such a pattern is $m-1$, then multiplying any number by $a^{m-1}$ is equivalent to multiplying it by 1. In the language of modular arithmetic, this can be stated $a^{m-1} \equiv 1 \pmod{m}$.

Fermat's Little Theorem, which we will not prove here, can be thought of as a generalization of this result that does not involve consideration of repeating patterns. More specifically:

**Theorem 20** (Fermat's Little Theorem)**.** *If $a$ is an integer and $p$ is a prime number that does not divide $a$, then $a^{p-1} \equiv 1 \pmod{p}$.*

You may have noticed the requirement that $p$ does not divide $a$. Why is this? To explain this, it pays to consider an example where $p$ does divide $a$. Consider what happens, for example, if $a = 20$ and $p = 5$. Of course $p = 5$ is a prime number, but it is also clear that $a^{p-1} \equiv 0 \pmod{p}$, since 5 evenly divides 20, and so there is never a remainder after dividing 20, or any power of it, by 5. So Fermat's Little Theorem can only consider cases where $p$ does not divide $a$.

**Example 1. Example 1.** What is the remainder of $50^{72}$ when divided by 73? Since 73 is a prime number, and since 50 is not a multiple of 73, then we have $50^{72} \equiv 1 \pmod{73}$. So the remainder of $50^{72}$ when divided by 73 is 1.

**Example 2.** What is the remainder of $100^{10}$ when it is divided by 11? Since 11 is a prime number, and since 100 is not a multiple of 11, then we have $100^{10} \equiv 1 \pmod{11}$. So the remainder of $100^{10}$ when divided by 11 is 1. Of course we can combine this congruence relation with itself (using Theorem 16) to obtain $100^{20} = 100^{10} \times 100^{10} \equiv 1 \times 1 = 1 \pmod{11}$. The same process can be repeated to show that $100^{30}$, $100^{40}$, etc, are also congruent to 1 mod 11.

**Example 3.** What is the remainder of $3^{49}$ when divided by 7? Fermat's Little Theorem tells us that $3^6 \equiv 1 \mod 7$, so we write $3^{50}$ in terms of $3^6$. We can write this as $3^{49} = 3 \cdot (3^6)^8$, which we can then reduce: $3 \cdot (3^6)^8 \equiv 3 \cdot 1^8 \equiv 3 \pmod{7}$.

**Example 4.** What is the remainder of $2^{432}$ when divided by 11? Of course 11 is a prime number, but the exponent here is not $p-1$, so how can we use Fermat's Little Theorem to help us? We can rewrite $2^{432}$ as $2^{430}2^2 = (2^{10})^{43}2^2$. Note that Fermat's Little Theorem tells us that $2^{10} \equiv 1 \mod 10$, which means that we can replace $2^{10}$ in this equation with 1. So we have $2^{432} = 2^{430}2^2 = (2^{10})^{43}2^2 \equiv 1^{43}2^2 \equiv 1 \cdot 2^2 \equiv 4 \pmod 1$1. Hence, the remainder of dividing $2^{432}$ by 11 is 4.

**Example 5.** What is $29^{25} \pmod{11}$? Fermat's Little Theorem tells us that $29^{10} \equiv 1 \pmod{11}$, so we want to rewrite $29^{25}$ as $29^{10} \cdot 29^{10} \cdot 29^5$. We then have $29^{25} \equiv 29^{10} \cdot 29^{10} \cdot 29^5 \equiv 1 \cdot 1 \cdot 29^5 \equiv 29^5 \pmod{11}$. Since $29 \equiv 7 \pmod{11}$, we can further simplify this to $7^5 = 7^2 \cdot 7^2 \cdot 7 \equiv 49 \cdot 49 \cdot 7 \equiv 5 \cdot 5 \cdot 7 \equiv 10 \pmod{11}$.

**Example 6.** What is $1^{10} + 2^{20} + 3^{30} + 4^{40} + 5^{50} + 6^{60} \pmod{11}$? Fermat's Little Theorem has $a^{10} \equiv 1 \pmod{11}$ for each term. Even when we take multiples of the exponent 10, we still have the same result. Therefore, each term contributes 1, and so the answer is the number of terms, 6.

Notice that each problem is different and requires thinking. Oftentimes, rewriting a large exponent as the product of smaller exponents can enable the use of patterns of Fermat's Little Theorem to further simplify a problem.

## 6.4 Diffie-Hellman Key Exchange

We can now use modular arithmetic to devise a secure communication protocol. We begin by discussing a method by which two people, far away from one another, can share a password that no one else can know. What is amazing is that both of them can send information publicly, yet end up with a mutually-shared password that only these two people know. How can they do that?

To motivate the general approach, consider the following dilemma. Suppose you and a friend would each like to paint your rooms with the same color. It's not important what color that is, but you want to make sure that no one else in town uses that color. How can you make this happen? If the two of you go to Lowe's or Home Depot together, you can choose a color, split the can, and go home. But suppose that you to live some distance away and won't have a chance to see each other. If one of you buys the paint and sends half of it to the other person, someone else, perhaps the delivery person, might intercept that color! Even if the person only sends the information about the paint, someone else might discover your scheme. Is there any way to solve this problem?

### Mixing Paints

It turns out that there is such a way, due to a very important "problem" that arises in mixing paints that some readers have likely encountered. Imagine going to the store and choosing a color you like, and also a bucket of white paint, which you use to make the color lighter. You go home and mix some blue paint and some white until you get the color that you think will look perfect. You begin painting the room but soon, after painting half of the walls, you realize that you didn't mix enough paint, and you'll need to make more. Now you have a mega-problem. You don't remember exactly how much white you added to the blue, and have no idea how to recreate the exact shade you made initially. Of course you can guess the proportions, but now there's a good chance that half of your walls will be one shade of blue, and the other half of the walls will be another shade. You've painted yourself into a figurative corner.

This problem highlights the following beautiful property of paints – it's very easy to mix them, but almost impossible to look at a mixed paint and determine how it was made. While this can be very frustrating for someone painting, this issue in fact allows you and your friend to solve your high-security room-painting needs. You can do the following. Each of you takes a gallon of white paint. Next, you take a colored paint of your choosing in an amount of your choosing and add that to the gallon of white paint you bought; you don't tell anyone how much you've added. Your friend does the same with whatever color they've chosen. Now each of you sends that paint to the other person. The important point to notice is that anyone that might see the paints in transit has no way of knowing what other paint you've mixed in and in what quantity. Perhaps you've added a quarter gallon of quarter gallon of Fountain blue, or perhaps it was a third of a gallon of Capri.

Now you have the paint your friend has mixed, and they have the paint that you've mixed. These paints are different, but you can now make them the same quite easily. Each of you adds to the paint in your hands the exact amount of whatever paint you've chosen and added to the other paint. The two paints are now identical and only the two of you have that color, since anyone in the middle who has seen the paint in transit has no way of determining what color and what amount each of you have added.

This beautiful "thought experiment" shows that it is possible for two people to work together to create information that is known only to them and secret from everyone else, even though they have shared some information publicly. This idea motivates the development of the Diffie-Hellman key-exchange protocol that is used regularly by computers when information must be sent securely. Of course computers do not send paints to one another, but through modular arithmetic they are able to achieve a similar result.

**Diffie-Hellman Key Exchange**

Alice and Bob would like to communicate securely. The Diffie-Hellman key exchange protocol allows them to work together to create a password that only the two of them will know, even while some of the information they exchange is completely public. To do this, they use numbers instead of paints. More specifically, they agree (publicly) on a modulus $m$ and an integer $g$, which is smaller than $m$ and which serves as their "white paint". Next, each of Alice and Bob chooses another secret number which they will share with nobody; we will use $a$ to refer to Alice's secret number and $b$ to refer to Bob's secret number.

Alice then calculates $g^a \pmod{m}$ and Bob calculates $g^b \pmod{m}$. Like with the paints, it is easy to create these numbers but almost impossible to figure out how they were made. That is, if you just know $g$, $m$, and $g^a \pmod{m}$, there is no known way of efficiently determining $a$. If $m$ is small we can use trial-and-error to quickly determine $a$, but in general the value of $m$ might have hundreds of digits (we're talking about numbers bigger than a trillion times a trillion times a trillion many times over). For this reason, Alice can send the number $g^a \pmod{m}$ to Bob and not worry that anyone will figure out her secretly chosen number $a$, even if they know $g$, $m$, and $g^a \pmod{m}$. Likewise, Bob can send over $g^b \pmod{m}$ and not worry that someone will figure out his secret number $b$. In this sense, they are sending over their specially-mixed paints and no one can figure out how they mixed them, even if they know that the base was white.

At this point Alice still remembers her secret number $a$ and now has a number $g^b \pmod{m}$ which she received from Bob. Using this, and modular exponentiation, she can quickly compute $(g^b)^a \pmod{m}$ by taking $g^b \pmod{m}$ to the $a^{th}$ power. Likewise, Bob still has his secret number $b$ and also knows $g^a \pmod{m}$, which Alice told him, allowing him to compute $(g^a)^b \pmod{m}$. We might remember from high-school algebra that $(x^a)^b = x^{ab} = x^{ba} = (x^b)^a$; the same rules hold in modular arithmetic, and so $(x^a)^b \equiv x^{ab} \equiv x^{ba} \equiv (x^b)^a \pmod{m}$. Therefore, Alice and Bob now have the same number $g^{ab} \pmod{m}$, and they are the only two people that know the number. Even though bad guys might know $g$ and $m$, and even $g^a$ and $g^b \pmod{m}$, they have no way to figure out $g^{ab} \pmod{m}$.

If Alice and Bob use $g = 3$ and modulus $m = 19$, for example, then if we can just compute $g^1, g^2, \ldots, g^{18}$ to determine all possible values of $g^a \pmod{m}$, and use that list to determine $a$ once we know $g^a \pmod{m}$. As noted above, however, $m$ is usually chosen to be a number with hundreds and hundreds of digits, and calculating a list of possible $g^a \pmod{m}$ values for every $a < m$ would take billions and billions of years, even if we had the most powerful computers in the world focused on that problem alone. It is thus the *practical* impossibility of determining $a$ that makes this protocol secure. We don't know whether one day someone will figure a way to determine $a$; if that happens, security as we know it will need new tools.

**NOTE**: In setting up this protocol, it is important to make "good" choices of $g$ and $m$. To highlight why some thought is necessary, consider choosing $g = 10$ and $m = 101$. We might notice quickly notice that $g^1, g^2, g^3, g^4, g^5 \ldots \equiv 10, 100, 91, 1, 10, \ldots \pmod{101}$. In other words, the repeating pattern has period 4, and so there are only 4 possible values of $g^n \pmod{101}$. That means that Alice and Bob will only be able to send over one of these 4 numbers, making it extremely easy to crack this code.

**Example 1.** Alice and Bob agree to use $g = 7$ and $m = 997$. Alice chooses $a = 5$ and Bob chooses $b = 10$; they keep these numbers secret, telling no one else about them. Alice then calculates $g^a = 7^5 \equiv 855 \pmod{m}$ and Bob calculates $g^b = 7^{10} \equiv 224 \pmod{m}$. Each sends their computed number to the other, so Alice receives 224 and Bob receives 855. They each then compute $g^{ab}$ (mod 997) by taking their received number and exponentiating it to their secret number. Alice determines that $224^5 \equiv 455 \pmod{997}$ and Bob determines that $855^{10} \equiv 455 \pmod{997}$. Both Alice and Bob now can use the number 455 as a shared secret password.

**Example 2.** Alice and Bob agree to use a base $g = 37$ and modulus $m = 2{,}305{,}843{,}009{,}213{,}693{,}951$. Alice chooses $a = 537$ and Bob chooses $b = 3024934$, which they use to calculate $g^a \pmod{m} \equiv 957{,}141{,}291{,}894{,}918{,}330$ and $g^b \pmod{m} = 2{,}210{,}741{,}389{,}954{,}762{,}204$. Each sends their computed number to the other, so Alice receives $g^b \pmod{m}$ and Bob receives $g^a \pmod{m}$. They each then compute $g^{ab} \pmod{m}$ by taking their received number and exponentiating it to their secret number. Alice determines $(g^b)^a \pmod{m} = 2{,}305{,}843{,}009{,}213{,}693{,}951$ and Bob determines that $(g^a)^b \pmod{m}$ is that same number. Alice and Bob now can use this number $g^{ab} \pmod{m}$ as a shared secret password to communicate securely. If a bad guy wanted to guess the secret numbers of Alice and Bob, they might need to perform millions of billions of computations to determine that $a = 537$ or that $b = 3024934$.

Of course calculating these numbers is not easy to do by hand, but can be done relatively easily by a well-programmed computer. This protocol allows computers to communicate with one another through insecure, public channels, yet protect secrecy of the transmitted information. It is used regularly by billions of electronic devices around the world every single day.

## 6.5 Random Numbers

The final topic we will cover in connection with modular arithmetic is that of random numbers, and how computers generate them. It turns out that there is an amazing, though quite mysterious, connection between modular arithmetic with large exponents and the generation of "random numbers". We put the words random numbers in quotations because in a sense that will become clear in a moment, the random numbers we generate will not really be random at all, even if a person looking at them doesn't really know that.

Random numbers appear regularly in our everyday lives in different forms. If you've ever watched an NFL ref flip a coin to determine how a game would start, or if you've ever rolled a die while playing a board game with friends, or if you've ever chosen a random card from a deck of cards, then you've witnessed the role that randomness can play in determining some course of events.

Computers too use randomness on a regular basis. If you've ever asked iTunes to shuffle your playlist, you have asked your computer to do something that is random. If you've ever played a video game, you've witnessed a computer make choices about many random details. If you've ever used a computer to create realistic-looking pictures, for a movie or a video game, the computer has likely needed to generate random numbers, to help mimic the randomness that appears so ubiquitous in nature. If you are trying to use a computer to simulate some physical, biological, chemical, economic process, chances are that you will need some randomness to make sure that your simulation is realistic. And finally, if you have ever communicated secure information through the internet, chances are that your computer has needed to generate some kind of random number.

Discussing the generation of random numbers by computers requires us to first consider the more general question of what we mean by random numbers. We begin with a very brief discussion of some basic ideas of probability.

### What are random numbers?

We begin with several simple exercises, to highlight three simple lessons of probability. As an exercise, think of a random number between 1 and 10. Imagine that you chose seven. Does that mean that seven is a random number? Is it more random than two or three or nine? Of course these are silly questions, and it doesn't make much sense to discuss whether individual numbers are random or not. The more interesting, and fruitful, question is whether some sequence of numbers is random. **Lesson 1**: There is no such thing as a random number. Instead we consider sets or sequences of numbers that are random.

Next, consider the following two sequences of numbers. Perhaps both sets of numbers correspond to a sequence of coin-flips, with heads indicated by 1's and tails indicated by 0's:

**a)** 0 1 0 1 0 1 0 1 0 1 0 1 0 1

**b)** 0 1 1 0 0 1 1 1 0 0 1 0 1 1

Sequence **a)** does not appear "random" – it's merely a repeating pattern of 0 and 1. What about the sequence **b)**? This one appears to be random, or at least much more so than the first. Now consider the following two sequences of integers.

**c)** 2 6 5 3 5 8 9 7 9 3 2 3

**d)** 7 5 2 3 10 4 6 9 8 1

Is the first set of numbers random? What about the second set? Looking back at **a)** and **b)**, these two appear significantly "more" random than those. Indeed, even if we thought that **b)** appeared random, we would admit that it appears random only when considering sequences of 0's and 1's, but not when considering more arbitrary sequences. If we we allow integers all the way up to 10, then neither of **a)** nor **b)** seem at all random. **Lesson 2**: Randomness is "relative". Whether or not a sequence of numbers is "random" depends on what numbers we are choosing from.

Finally, consider the following set of numbers, chosen from 5 to 30.

**e)** 20 19 14 12 19 18 17 19 20 14 18 12 20

Are these numbers random? Even though you might believe that these numbers are all from between 5 and 30, they certainly don't seem very random, as all of them are greater than 10 and no larger than 20. Could these numbers have indeed been chosen from between 5 and 30? Indeed these numbers were chosen from between 5 and 30, yet the manner in which they were chosen was not *uniform*. That is, there was not an equal chance that 5 and 15 and 25 were chosen. In fact, these numbers were obtained as follows. To obtain each number, I rolled a die five times and added the sum of their values. Since there were five dice, of course the minimum value I could get was 5 (if I rolled only 1's) and the maximum was 30 (if I rolled only 6's). But it's much easier to obtain numbers in the middle than numbers at the extremes. **Lesson 3**: Randomness does not need to be *uniform*; the probability of choosing one object can be different from the probability of choosing another one.

Along these lines, consider the following "random" SAT scores:

**f)** 1590 1470 2100 830 1930 2040 840 2050 1950 840 640.

Of course these numbers must be between 600 and 2400, and must be divisible by 10. But despite being among certain values, they don't of course, represent a "random" sample, certainly not among Penn students, but not even among the general population. Therefore, when choosing numbers randomly, we must always specify the probability of each outcome. Of course, many more students score a 1700 than score an 800 or 2300.

Lessons 2 and 3 highlight the need for describing a probability distribution before considering whether a particular set of numbers is random or not – randomness cannot be sensibly discussed without this kind of frame of reference. A probability distribution describes that frame of reference by giving us a list of possible values that can be chosen, and the probabilities of choosing each of those values. In all of our examples, we have considered distributions with a finite number of possible choices, though in theory we can consider more complicated ones.

In discussing how computers generate random numbers, we will only consider a **uniform distribution** on a finite set of numbers. In particular, suppose we want a computer to choose a number between 1 and 100 with a one percent chance of choosing any particular number. More generally, we might want to ask a computer to choose a number between 1 and $N$, with an equal chance of choosing any one of the $N$ choices. How can a computer do this?

## How Computers Generate Random Numbers

Computers are very, very, very good at completing certain tasks. For example, computers are very good at adding numbers. Or multiplying them. Or dividing them and taking square roots and exponents. And they can do this all really, really, really well, and much faster and more accurately than any of us. But they can only do what you tell them to do, with specific instructions. This raises the question of how can a computer generate a random number? How can it generate something random by following a fixed set of rules? The honest truth is that it can't. A deterministic machine that just follows orders, and follows fixed instructions, cannot generate true random numbers. However, we will see that, using modular arithmetic, a computer *can* generate numbers that appear random for most intents and purposes.

**Irrational numbers.** One source of random numbers can be found in the decimal expansion of irrational numbers. Consider, for example, the decimal expansions of $\pi$, $e$, and $\sqrt{2}$:

**g)** (3.) 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6

**h)** (2.) 7 1 8 2 8 1 8 2 8 4 5 9 0 4 5 2 3 5 3 6

**i)** (1.) 4 1 4 2 1 3 5 6 2 3 7 3 0 9 5 0 4 8 8 0

It is commonly believed that these numbers are very random, in a way that can be made precise. In practice, people have looked at the first billion digits of $\pi$ and the digits seem to appear randomly distributed. For example, roughly one tenth of all digits are 0's, one tenth are 1's, and so forth. No one, though, knows whether this is true indefinitely. As far as we know, after the first trillion digits, there tend to be considerably fewer 3's than any other digit; it is also possible that there are no 7's appearing after the first fifty trillion digits. No one knows how to prove anything about this.

In practice, irrational numbers are not commonly used to generate random-looking numbers for several reasons. First, generating digits in this manner is expense, both in terms of computational time and memory. Furthermore the resulting numbers are very predictable. If I can figure out that you are using $\pi$ or $e$ or $\sqrt{2}$ to generate your random numbers, then in theory I can exactly predict every number that you will ever create. We will see later that this doesn't have to be a fatal flaw, but in practice, it often is.

## Linear congruent generators

The most widely-used random number generators are called linear congruent generators, and in this section we will learn about what they are and how they are used. Remember that computers can follow orders, so we are trying to find directions that create numbers that appear random.

The simplest type of random congruent generator is a sequence of numbers of the form:

$$s, sa^1, sa^2, sa^3, sa^4, \dots \pmod{m}, \tag{75}$$

where $a$ is called the multiplier, $m$ is the modulus, and the first element $s$ is called the "seed". Each term in the sequence can be obtained by multiplying the term before it by $a$, and then taking it mod $m$ (i.e., the remainder after dividing it by $m$). We can rewrite the above in a slightly more condensed form. In particular, if we use $x_i$ ot indicate the $i^{th}$ number in the sequence, we can write:

$$x_i = a \cdot x_{i-1} \pmod{m}, \tag{76}$$

where we let $x_0 = s$. This definition defines each term as the product of $a$ and the preceding term in the sequence.

Let us consider a simple example. We choose a seed $s = 1$, a multiplier $a = 7$, and a modulus $m = 11$. These choices of $s$, $a$, and $m$ give us a sequence:

$$1, 7, 5, 2, 3, 10, 4, 6, 9, 8, 1, 7, 5, \dots \tag{77}$$

These numbers look fairly random, but the problem is that they repeat too quickly. For reasons we have already discussed, the length of this sequence can be no longer than $m-1$. Therefore, in practice, random number generators try to use a large modulus $m$.

Next we consider a much larger modulus $m = 2^{31}$ and multiplier $a = 65539$; this was the basis for an historically-important random number generator developed and used by IBM in the 1960's. If we choose a seed $s = 123456789$, then our first several numbers are:

$$123456789, 1663592255, 280507837, 1743102263, 1491592101, \dots \tag{78}$$

At first glance these numbers might appear fairly "random", and indeed they are evenly distributed between 1 and $2^{31}$. However, you might notice that every term is odd, which occurs when the seed $s$ is chosen odd; if $s$ is chosen even, then every subsequent term will be even. Although we can get around this problem by always dropping the last digit, this problem highlights some of the challenges involved in designing random number generators.

In practice, however, linear congruent generators are the most widely-used pseudo-random number generators in common use, and much work has been put into determining good choices of multiplier $a$, modulus $m$, and seed $s$.

A slight generalization of the example described here involves not only multiplying preceding terms by a constant $a$, but also adding a number to it. More concretely, we choose a constant integer $c$ which we call the *increment* and add that after multiplying the previous term by $a$. In equation form,

$$x_i = a \cdot x_{i-1} + c \pmod{m}. \tag{79}$$

To see how this works, let's consider the simple example we considered before, where we chose a multiplier $a = 7$, a modulus $m = 11$, and a seed $s = 1$. Let us now also choose an increment $c = 5$. Instead of sequence in (77), we now get:

$$1, 10, 7, 8, 4, 9, 0, 3, 2, 6, 1, 10, 7, \ldots \tag{80}$$

The pattern is again periodic, but the order of the numbers have changed. In some situations, including the extra incremental term $c$ can improve certain properties of the random numbers generated, but sometimes it can make things much worse. Consider, for example, what happens in the previous example if we instead use an increment $c = 5$. Notice that $(7 \cdot 1 + 5) = 12 \equiv 1 \pmod{11}$. In words, if we begin with 1, multiply it by 7 and add 5, and then consider the remainder after dividing by 11, then the remainder is 1. Therefore the "random" sequence generated will end up being an endless sequence of 1's – not a very random sequence at all!

## Randomness testing

We have already seen some potential pitfalls in the development of random number generators. Sometimes the period of the repeating pattern is very short. Other times the pattern might be quite long but the generated numbers are all odd, or all even. Even if we are content with using deterministic algorithms to generate number sequences that only look random, we still want to make sure that they indeed appear random. Over the last fifty years, many tests have been developed to determine whether a particular set of pseudo-randomly-generated numbers indeed appear random. One of the best-known and most powerful such tests is called the spectral test.

To help understand the spectral test, consider the following sequence of numbers generated by linear congruence generators with seed $s = 7$, multiplier $a = 5$, and modulus $m = 97$:

$$7, 35, 78, 2, 10, 50, 56, 86, 42, 16, 80, 12, 60, 9, 45, 31, \ldots \tag{81}$$

Next consider the sequence of numbers when we keep the modulus and seed, but change the multiplier $a$ to 10:

$$7, 70, 21, 16, 63, 48, 92, 47, 82, 44, 52, 35, 59, 8, 80, \ldots \tag{82}$$

The two sequences appear at first to be equally "random". However, consider what happens if we plot all pairs of consecutive terms in each sequence. For the first sequence, for example, we would plot points $(7, 35)$, $(35, 78)$, $(78, 2)$, etc.

For the second sequence, we plot $(7, 70)$, $(70, 21)$, $(21, 16)$, etc. Something mysterious occurs the quickly highlights the different strengths of the two sequences. When we used the multiplier $a = 5$, all of the 96 points are "bunched up" on

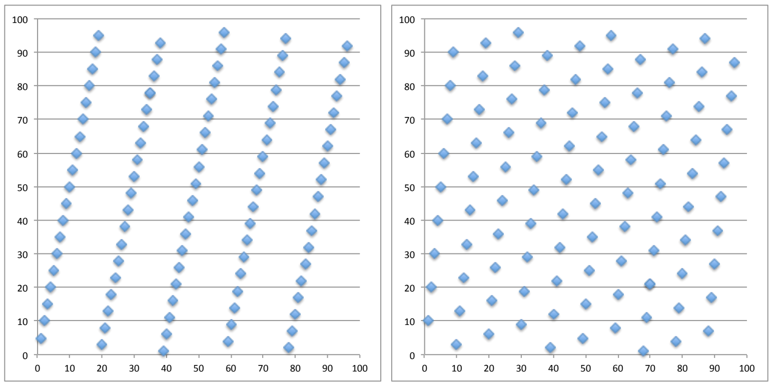

Figure 37: Linear congruence generators using modulus $m = 97$ and initial seed $s = 7$ The points on the left are taken from a sequence generated using a multiplier $a = 5$; those on the right are taken from a sequence generated using a multiplier $a = 10$.

five lines, where when we use the multiplier $a = 10$, the 96 generated points are much more spread out. A branch of mathematics called Fourier analysis can be used to make these ideas precise, and in practice this kind of test is used to determine whether particular choices of multiplier and modulus are random "enough" for particular applications.