Stanford SPLASH

Showing off with (Math) bases

Fay' Ibrahima

Institute for Computational & Mathematical Engineering,
Stanford University

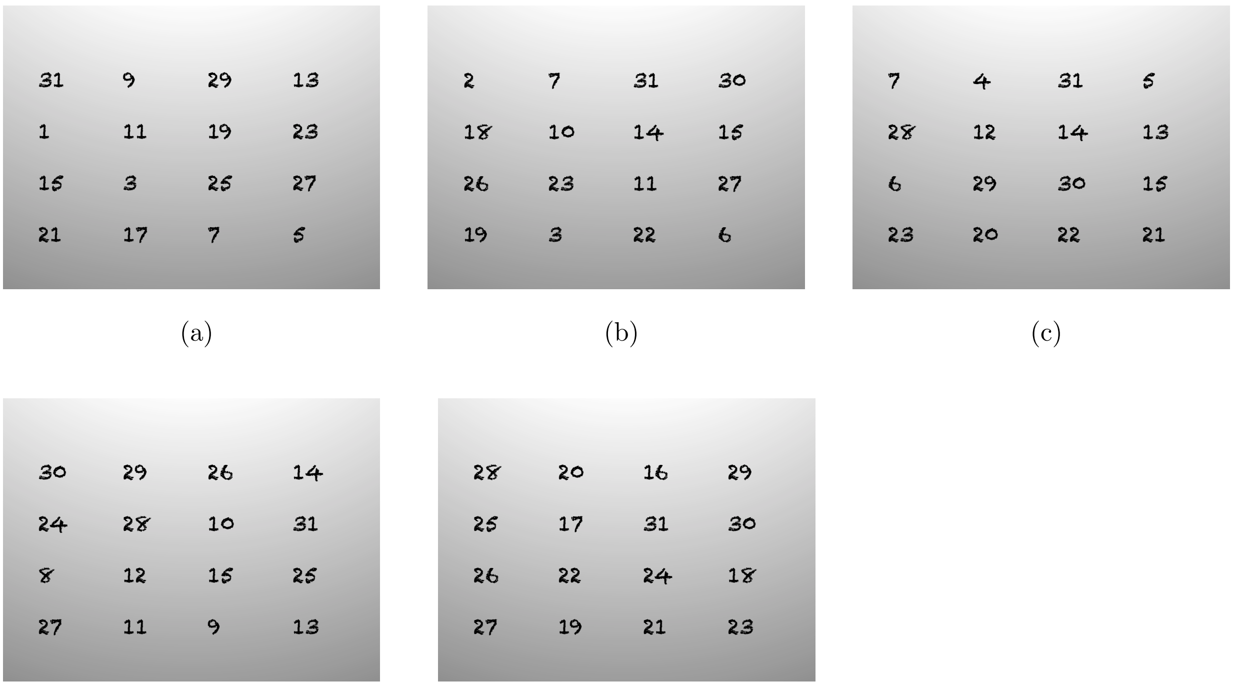
November 7, 2015

1 Introduction

Mathematics are a very broad and rich field of science. What I like about Maths is that it can be very applied while relying on very deep and interesting concepts. In this session, we explore two tricks that require the use of the representation of numbers in a certain base. In the first example, we modify the base to compress the information about the number. In the second example, we exploit the decimal base to discover rules about divisibility.

2 I will guess your number (yes/no)

Pick a number between 1 and 31. I have 5 cards, shown in Figure 1. I will guess your number by only asking you if the number shows up in each card. Now you might be like “Hold on, he only has 5 cards and he can find out a number out of 31. How is it possible???”. The short answer: magic. The long answer: Maths.



(a) (b) (c) (d) (e)

Figure 1: 5 cards for the trick (yup, I’ve guessed your number).

2.1 The card trick explained

The trick is that you can write a number in different bases (see next section for quick explanation). The decimal system has nothing special. When someone says "yes" or "no", that person is actually giving away the number... in base 2! Also know as the binary base. "yes" means 1 and "no" means 0. Let's take an example. Suppose that the number to guess is 13. Let's mimic the conversation (in parenthesis, how I do the calculation in my head).

- Fay, showing the first card: "Is it here?"
- Student: "Yes."
- Fay, showing the second card: "(OK. that's 1) Is it here?" [The number in binary is ????1]
- Student: "No."
- Fay, showing the third card: "(OK, the number is $1 + 0 \times 2 = 1$) Is it here?" [The number in binary is ???01]
- Student: "Yes."
- Fay, showing the fourth card: "(OK, the number is $1+0\times2+1\times4 = 5$) Is it here?" [The number in binary is ??101]
- Student: "Yes."
- Fay, showing the fifth card: "(OK, the number is $1+0\times2+1\times4+1\times8 = 13$) Is it here?" [The number in binary is ?1101]
- Student: "No."
- Fay, showing off: "(OK, the number is $1+0\times2+1\times4+1\times8+0\times16 = 13$) Your number is 13" [The number in binary is 01101]

What's special with the numbers in the first card (a) is that they are all represented in binary by: `xxxx1`, where `x` is either 0 or 1 (in particular, 1 is only in (a)). The numbers in (b) are all represented by: `xxx1x` (in particular, 2 is only in (b)); the numbers in (c) by: `xx1xx` (in particular, 4 is only in (c)); the numbers in (d) by: `x1xxx` (in particular, 8 is only in (d)) and the numbers in (e) by: `1xxxx` (in particular, 16 is only in (e)).
Well, the question now is what is binary representation? And more generally, what is a base?

2.2 Numbers and Bases

Usually, we right numbers with digits going from 0 to 9. But if you think about it, this 10 digits are arbitrary (and are convenient because we have 10 fingers). For instance, when your watch has different ways to represent time. The hours repeat every 12 or 24 hours (depending on how you read time) and you have 60 minutes in an hour. You are using a different way to represent numbers associated with time! Let's illustrate how the decimal system works. Let's consider the number 136778_{10} (the subscript is there to emphasize that we represent the number in base 10). This is how we can read it:

1	3	6	7	7	8	digits
10^5	10^4	10^3	10^2	10^1	10^0	base powers

Or we can write it as:

$$\begin{aligned} 136778 &= 1 \times 10^5 + 3 \times 10^4 + 6 \times 10^3 + 7 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 \\ &= 1 \times 100000 + 3 \times 10000 + 6 \times 1000 + 7 \times 100 + 7 \times 10 + 8 \times 1 \end{aligned} \tag{1}$$

Note. We recall that $10^k = \underbrace{10 \times 10 \times \cdots \times 10}_{k \text{ times}}$. For instance, $10^4 = 10 \times 10 \times 10 \times 10$. You can replace 10 by any number, it works the same way.

Now let's illustrate how the binary system works. Let's consider the number 101101_2 (in base 2). Here is how we can read it:

1	0	1	1	0	1	digits
2^5	2^4	2^3	2^2	2^1	2^0	base powers

Or we can write it as:

$$\begin{aligned} 101101_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\ &= 45_{10} \end{aligned} \tag{2}$$

More generally, in base b, we have $b - 1$ digits (they can even be letters, or whatever you want! cf. base 16 [a.k.a. hexadecimal, a.k.a. hex].) and work with powers of b: b^k, with $k = 0, 1, 2, \ldots$

Exercises.

1. Prove that an integer can be written in any base
2. Think of a more innovative way to deliver (or design) the binary card trick

2.3 Card trick in base 3?

If $2^4 = 16$, which means that we can guess 16 numbers with 4 cards (including 0), $3^4 = 81$. Therefore, with 4 cards, and using base 3, we should be able to guess 81 numbers! The question is: how to design a magic card trick in base 3? The issue now is that we have 3 digits per level: 0, 1 and 2. And while in binary the interpretation was clear, in base 3 it is not trivial (yes/no/maybe? :p).
One way to go around it is to proceed as follow: for each card, form 2 columns and ask if number in column 1 or 2 or none (0). Then the number is naturally expressed in base 3.

<table>
<tr><td>1</td><td>2</td><td>1</td><td>2</td><td>1</td><td>2</td><td>1</td><td>2</td></tr>
<tr><td colspan="2">or 0</td><td colspan="2">or 0</td><td colspan="2">or 0</td><td colspan="2">or 0</td></tr>
<tr><td colspan="2">3^3</td><td colspan="2">3^2</td><td colspan="2">3^1</td><td colspan="2">3^0</td></tr>
</table>

Table 1: Example of a base 3 card trick setup.

Exercise.

1. Can you think of a better way to implement the base 3 card trick?

2.4 Things to think about

The exciting thing about the binary card trick is that you can compress information about numbers in sequence of 0's and 1's. For instance, with 10 cards, you can guess numbers from 0 to $2^{10} - 1 = 1023$. With 20 cards, you can go to $2^{20} - 1 = 1048575$. It's huge! The caveat, though, is that you would need very very very large cards (and a long long patience) to write and read all these numbers...
Another remark is that, in the end, to do calculations, you only need to know two states: “yes” or “no”, 1 or 0, open or close, etc. Hence, you can use electrical signals to calculate. For instance, you record a signal every short period of time. If you record something about a threshold, then it is considered to be 1 if not 0. This is essentially how computer started, and

why today we have supercomputers. Binary representation is essential in Computer Science.

3 Divisibility rules: how?

3.1 Divisibility rules

Now let's switch gears (, go back to the good old base 10) and look at the problem of knowing whereas a number is a multiple of another one. For $n \leq 11$, we have simple and commonly known divisibility rules. An integer m is a multiple of

- 2 if its last digit is a multiple of 2, that is, even: 0, 2, 4, 6, 8,
- 3 if the sum of its digits is a multiple of 3, (and you can apply the rule recursively)
- 4 if the two last digits form a multiple of 4,
- 5 if the last digit is a multiple of 5, that is: 0 or 5,
- 6 if it obeys simultaneously the rules of 2 and 3,
- 7 if the subtraction of twice the last digit to the rest is a multiple of 7 (can apply recursively)
- 8 if the addition of the last digit to twice the rest is a multiple of 8 (can apply recursively)
- 9 if the sum of all the digits is a multiple of 9 (can apply recursively)
- 10 if it ends with 0,
- 11 if the alternating sum of the digits is a multiple of 11 (can apply recursively)

We can see that we have very different rules. Some look simple (2, 3, 9, 10), while some look out of the blue (how the hell do we get the 7 rule?). Let's try to understand how to come up with these divisibility rules. Obviously, they will require the use of base (10).

3.2 Building divisibility rules: modulo calculus

To be able to obtain divisibility rule, we are going to rely on this assertion: a number n is divisible by (a multiple of) m if its remainder by division by m is 0. In other words, there exists an integer k such that $n = m \times k + 0$. For instance, 51 is divisible by 17 because $51 = 17 \times 3 + 0$, while 33 is not divisible by 7 because $33 = 7 \times 4 + 5$.
Now it is worth noticing that a remainder of a number is never higher than this number. More precisely, if r is a remainder of m, then $0 \leq r < m$. When we are checking of a number is a multiple of another, we only care about the remainder, not the actual number, which motivates the use of modulus calculus. Modulus calculus is simply a sophisticated name to say analysis of the remainder (modulo a number). For instance modulo 7, 33 is the same thing as 5. And modulo 17, 51 is the same thing as 0. We usually write $33 \equiv 5 \mod 7$. Let's call 5 a representation of 33 modulo 7. We have some desirable properties for representations:

- Modulo a prescribed integer, the representation of the sum of two numbers is the sum of the representations. That is if $m_1 \equiv r_1 \mod n$ and $m_2 \equiv r_2 \mod n$, then $m_1 + m_2 \equiv r_1 + r_2 \mod n$. (Prove it)

- Modulo a prescribed integer, the representation of the product of two numbers is the product of the representations. That is if $m_1 \equiv r_1 \mod n$ and $m_2 \equiv r_2 \mod n$, then $m_1 \times m_2 \equiv r_1 \times r_2 \mod n$. (Prove it)

Armed with these two properties and the knowledge of decimal base, we can unveil the secret of the divisibility rule for all the above. For instance, let's look at the divisibility rule for 3. Let's consider a number m with representation in base 10,

$$m = a_0 \times 10^0 + a_1 \times 10^1 + \cdots + a_p \times 10^p \tag{3}$$

We can see that $10 = 3 \times 3 + 1$, so that with our new formulation, $10 \equiv 1 \mod 3$. Now, by the multiplication property, $10^2 = 10 \times 10 \equiv 1 \times 1 \mod 3$, i.e., $10^2 \equiv 1 \mod 3$. Again, $10^3 = 10 \times 10^2 \equiv 1 \times 1 \mod 3$, which means $10^3 \equiv 1 \mod 3$. And we can see by recursion that for all $p \geq 0$ integer, $10^p \equiv 1 \mod 3$.
Therefore, by additivity and multiplication properties, $m \equiv a_0 + a_1 + \cdots + a_p \mod 3$. Hence m is a multiple of 3 if and only if $a_0 + a_1 + \cdots + a_p$ is a multiple of 3.

3.3 How the hell do we get the 7 rule?

Let's consider a number m with representation in base 10,

$$\begin{aligned} m &= a_0 \times 10^0 + a_1 \times 10^1 + \cdots + a_p \times 10^p \\ &= a_0 + 10 \times \left(a_1 \times 10^0 + \cdots + a_p \times 10^{p-1}\right), \end{aligned} \tag{4}$$

where $p \geq 1$ is an integer. We can see that in parenthesis, we have a number consisting of all the digit of m except for the last one (the unit digit). Let's call this number q. Since $10 \equiv 3 \mod 7$, we have $m \equiv a_0 + 3 \times q \mod 7$, hence $2 \times m \equiv 2 \times a_0 + 6 \times q \mod 7$. But $6 = 7 - 1$, so that $2 \times m \equiv 2 \times a_0 - q \mod 7$. Finally, since 7 is odd, if m is a multiple of 7, so is $2 \times m$. Therefore, m is a multiple of 7 is equivalent to $2 \times a_0 - q$ multiple of 7.

Exercise.

1. Prove the divisibility rule for 8.
2. Is there a pattern in the simplicity of the divisibility rule as a function of the number? (Wikipedia enumerates quite a few of them. You can take a look at them and try to see if there is a trend.)

3.4 Things to think about

We touched here a little bit of number theory, as well as (remotely) Ring and group theory. The study of divisibility is primordial in cryptography, and you can learn more by being curious about RSA code, prime numbers and internet security.

4 Conclusion

The study of Maths have a significant impact in most of the engineering fields (and is full of wonders by itself). Not only is Maths useful, but it can also be enjoyable. Don't hesitate to try the card trick at home! I can guarantee that even adults might not know how it works. Just reply: "It's magic!".