Binary Numbers Magic Trick

Math Circle

April 07 2013

Can you read people’s mind?

- Pick a number from 1 to 63
- If your number appear in the card say yes, otherwise say no

- Let’s guess which number you picked

What is the trick?

- Let’s start with a simpler case:
- Pick a number from 1 to 3

2 3	1 3

- Can you guess the trick?

A little more complicated case

- Pick a number from 1 to 7

4 5 6 7	2 3 6 7	1 3 5 7

Suppose you can only use 0 and 1 as your digits

- Suppose you can only use 0 and 1 in your number system. Can you express every number in terms of 0 and 1 only?
- Our ordinary numbers are called " base-10 number system" or "decimal" because….

$$798 = (7 \times 100) + (9 \times 10) + (8 \times 1)$$

$$= (7 \times 10 \times 10) + (9 \times 10) + (8 \times 1)$$

$$= (7 \times 10^2) + (9 \times 10^1) + (8 \times 10^0)$$

Note that $10^0 = 1$, $(\text{any non-zero number})^0 = 1$

Suppose you can only use 0 and 1

- In base-10 number system, digits are… 0,1,2,3,4,5,6,7,8,9

- In base- 2 number system, digits are…

  7 = ? x 2 x 2 + ? x 2 + ? X 1

- In base- 3 number system, digits are…

  7 = ? x 3 + ? x 1

Binary Numbers

- You can even have base – 12 number system (with your own creation of 2 more symbols)! However, base – 2 number system is particularly simple and useful as it is used in computer systems. Since there are only two modes - 0 (yes, on) and 1 (no, off) – this can be easily stored. Of course, every decimal number can be represented in a binary form.

Binary Numbers

- A binary number is often written as:

$(110)_2 = 1 \times 2 \times 2 + 1 \times 2 + 0 \times 1 = ?$

Compare with

$110 = 1 \times 10 \times 10 + 1 \times 10 + 0 \times 1 = 110$

In fact, 110 in binary system is $(1101110)_2$

So, how we convert a base-10 number into a binary number?

Let's try it! I need some volunteers!

So… how does this relate to our magic square?

- Recall…

2	**1**
3	3

- Suppose you choose 2. Then you will say yes to the first square and no to the second square. Then yes translates to 1 and no translates to 0. Therefore, your number turns out to be:

$$1 \times 2 + 0 \times 1 = 2 = (??)_2$$

This also is the number you get by simply adding the first number of "yes" square(s).

A little more complicated…

4 5 6 7	2 3 6 7	1 3 5 7

- Suppose you choose 5.
- Then you will say yes(1), no(0), yes(1)

Which means…

$1 \times 4 + 0 \times 2 + 1 \times 1 = 5 = (???)_2$

Again, this is the number you get by adding the first number of "yes" squares.

Q. What is so special about the first number in each square?

Let's make our own magic squares!

- The first number in each square is a power of 2: $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16,\ldots,2^{10}=1024$ (is this a familiar number?).
- Yes or no corresponds to if your n^{th} digit is 1 or 0. So if your number is $10 = (1010)_2$, then you want to make sure 10 is in both 8-square and in 2- square. Of course, 10 should not be in other squares. How about 11?

Let's make mind-reading square for 1 – 15!

- First is to convert every numbers from 1 to 15 into binary numbers.
- Then put each number into "yes" squares according the rule

Let’s check:

1	9
3	11
5	13
7	15

2	10
3	11
6	14
7	15

4	12
5	13
6	14
7	15

8	12
9	13
10	14
11	15

Now it's your turn to make the mind-reading square for 1-31! When you are done, play with your friends to make sure it is working.

Is this look familiar?

- Compare your magic card with:

1 3 5 7	2 3 6 7	4 5 6 7	8 9 10 11	16 17 18 19
9 11 13	10 11 14	12 13 14	12 13 14	20 21 22
15 17 19	15 18 19	15 20 21	15 24 25	23 24 25
21 23 25	22 23 26	22 23 28	26 27 28	26 27 28
27 29 31	27 30 31	29 30 31	29 30 31	29 30 31

An algorithm to convert numbers faster!

- How can be convert a decimal number into a base-n number?
- How can we convert a number so large such as 209203 to a binary number efficiently?

Binary Arithmetic

- Just like base-10 numbers, binary numbers also can be added, subtracted, divided and multiplied. How can we do this?
- What is $(0.001)_2$ in decimal? Can you convert a number such as 2.45, 1/3 into a binary number?
- Why is there always equal number of binary numbers in each table?