Data

Dr. Mihail

August 21, 2018

Data

How is data stored in a computer?

10101000101010100101010010101

Number systems

Bases

What is a number base?

Number systems

Bases

What is a number base?

Definition: the number of digits used in a numbering system.

Notation

Typically, when we write a number that is not a **natural** base 10 number, we write it as:

$$N_b$$

where N is the number and b is the base.
Examples: 1001_2, 343_8, $-FF12_{16}$

Counting

For the remainder of this course, we will assume 0 is a number.

Counting

For the remainder of this course, we will assume 0 is a number.

Let's count in base 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,... what happens next?

Counting

For the remainder of this course, we will assume 0 is a number.

Let's count in base 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,... what happens next?

How many digits did we use?

Counting

For the remainder of this course, we will assume 0 is a number.

Let's count in base 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,... what happens next?
How many digits did we use?

Add a digit

The number of digits is increased by one, so now we have two digit numbers: 10, 11, 12, ..., 97, 98, 99,... now what?

Counting

For the remainder of this course, we will assume 0 is a number.

Let's count in base 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,... what happens next?
How many digits did we use?

Add a digit

The number of digits is increased by one, so now we have two digit numbers: 10, 11, 12, ..., 97, 98, 99,... now what?

We add another digit, so now we have three digit numbers: 100, 101, 102, ..., 997, 998, 999, ...

Counting

For the remainder of this course, we will assume 0 is a number.

Let's count in base 10

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,... what happens next?
How many digits did we use?

Add a digit

The number of digits is increased by one, so now we have two digit numbers: 10, 11, 12, ..., 97, 98, 99,... now what?

We add another digit, so now we have three digit numbers: 100, 101, 102, ..., 997, 998, 999, ...

Pattern

Do you remember when you first learned how to count?

Digit nomenclature

Example

Take the number 512 in base 10.
The digits 2, 1 and 5 have a special meaning: 2 units, 1 ten and 5 hundreds.

More generally

- Units: 10^0
- Tens: 10^1
- Hundreds: 10^2
- Thousands: 10^3
- Tens of thousands: 10^4
- Hundreds of thousands: 10^5
- Millions: 10^6
- ...etc.

Rewriting numbers in base 10

Examples:

$$512 = 2 * 10^0 + 1 * 10^1 + 5 * 10^2$$

Rewriting numbers in base 10

Examples:

$$512 = 2 * 10^0 + 1 * 10^1 + 5 * 10^2$$
$$4393 = 3 * 10^0 + 9 * 10^1 + 3 * 10^2 + 4 * 10^3$$

Rewriting numbers in base 10

Examples:

$$512 = 2 * 10^0 + 1 * 10^1 + 5 * 10^2$$
$$4393 = 3 * 10^0 + 9 * 10^1 + 3 * 10^2 + 4 * 10^3$$
$$43058 = 8 * 10^0 + 5 * 10^1 + 0 * 10^2 + 3 * 10^3 + 4 * 10^4$$

Rewriting numbers in base 10

Base

$$512 = 2 * 10^0 + 1 * 10^1 + 5 * 10^2$$
$$4393 = 3 * 10^0 + 9 * 10^1 + 3 * 10^2 + 4 * 10^3$$
$$43058 = 8 * 10^0 + 5 * 10^1 + 0 * 10^2 + 3 * 10^3 + 4 * 10^4$$

Digit position and base exponent

$$5_2 1_1 2_0 = 2 * 10^0 + 1 * 10^1 + 5 * 10^2$$
$$4_3 3_2 9_1 3_0 = 3 * 10^0 + 9 * 10^1 + 3 * 10^2 + 4 * 10^3$$
$$4_4 3_3 0_2 5_1 8_0 = 8 * 10^0 + 5 * 10^1 + 0 * 10^2 + 3 * 10^3 + 4 * 10^4$$

Relating number of digits with quantity of numbers we can represent

How many numbers can we represent with x digits in base 10?
Example: 457 has 3 digits. 3 digit numbers have the form abc, where $a, b, c \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Answer:

Relating number of digits with quantity of numbers we can represent

How many numbers can we represent with x digits in base 10?
Example: 457 has 3 digits. 3 digit numbers have the form abc, where $a, b, c \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Answer:

$$10^x$$

Other bases

What if our digit "vocabulary" was a set of size 2?

We only have 0 and 1.

Can we count?

Other bases

What if our digit "vocabulary" was a set of size 2?

We only have 0 and 1.

Can we count?

Let's try

0, 1, ... now what?

Other bases

What if our digit "vocabulary" was a set of size 2?

We only have 0 and 1.
Can we count?

Let's try

0, 1, ... now what?
Add another digit: 00, 01, 10, 11,...
We have a total of 4 numbers. What next?

Other bases

What if our digit "vocabulary" was a set of size 2?

We only have 0 and 1.

Can we count?

Let's try

0, 1, ... now what?

Add another digit: 00, 01, 10, 11,...

We have a total of 4 numbers. What next?

Add another digit: 000, 001, 010 ,011, 100, 101, 110, 111.

We now have a total of 8 numbers. How many numbers can 4 digits represent?

Other bases

What if our digit "vocabulary" was a set of size 2?

We only have 0 and 1.

Can we count?

Let's try

0, 1, ... now what?

Add another digit: 00, 01, 10, 11,...

We have a total of 4 numbers. What next?

Add another digit: 000, 001, 010 ,011, 100, 101, 110, 111.

We now have a total of 8 numbers. How many numbers can 4 digits represent?

$2^4 = 16$

Counting in base 2

Counting

Base 10	Base 2
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
...	...

Rewriting numbers in base 2

Same principle as base 10

Let's look at a base 2 number with 4 digits. If $a, b, c, d \in \{0, 1\}$ are the digits, then:

$$abcd_2 = d * 2^0 + c * 2^1 + b * 2^2 + a * 2^3$$

Rewriting numbers in base 2

Same principle as base 10

Let's look at a base 2 number with 4 digits. If $a, b, c, d \in \{0, 1\}$ are the digits, then:

$$abcd_2 = d * 2^0 + c * 2^1 + b * 2^2 + a * 2^3$$

Example: 1011_2

$$1011_2 = 1 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 = 1 + 2 + 8 = 10_{10}$$

Rewriting numbers in base 2

Same principle as base 10

Let's look at a base 2 number with 4 digits. If $a, b, c, d \in \{0, 1\}$ are the digits, then:

$$abcd_2 = d * 2^0 + c * 2^1 + b * 2^2 + a * 2^3$$

Example: 1011_2

$$1011_2 = 1 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 = 1 + 2 + 8 = 10_{10}$$

Conversion

We just learned how to convert a base 2 (binary) number to base 10 (decimal). What about decimal to binary?

Take a step back

Question:
In a base 10 number, how can we find the units, tens, hundreds, etc.?

Take a step back

Question:
In a base 10 number, how can we find the units, tens, hundreds, etc.?
Naïve answer:
Duh! just look at the position from right to left. Example: in 456, there are 6 units, 5 tens, 4 hundreds, etc.

Modulo

For base 10 numbers, the remainder of the division by 10 operation (called **modulo** operator) is easy to compute.
Example: what is the remainder of the division by 10 operation for 456?

Take a step back

Question:
In a base 10 number, how can we find the units, tens, hundreds, etc.?
Naïve answer:
Duh! just look at the position from right to left. Example: in 456, there are 6 units, 5 tens, 4 hundreds, etc.

Modulo

For base 10 numbers, the remainder of the division by 10 operation (called **modulo** operator) is easy to compute.
Example: what is the remainder of the division by 10 operation for 456?
Answer: 6. We just found the number of units.

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save $x \bmod 10$ (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:

Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
    Save x mod 10 (remainder of the division by 10)
    Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: x mod $10 = 5$. Save 5 (tens)

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: x mod $10 = 5$. Save 5 (tens)
Step 4: $x = \frac{x}{10} = \frac{45}{10} = 4$

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: $x \bmod 10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: $x \bmod 10 = 5$. Save 5 (tens)
Step 4: $x = \frac{x}{10} = \frac{45}{10} = 4$
Is $x > 0$? Yes. Continue.

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: x mod $10 = 5$. Save 5 (tens)
Step 4: $x = \frac{x}{10} = \frac{45}{10} = 4$
Is $x > 0$? Yes. Continue.
Step 5: x mod $10 = 4$. Save 4 (hundreds)

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: x mod $10 = 5$. Save 5 (tens)
Step 4: $x = \frac{x}{10} = \frac{45}{10} = 4$
Is $x > 0$? Yes. Continue.
Step 5: x mod $10 = 4$. Save 4 (hundreds)
Step 6: $x = \frac{x}{10} = \frac{4}{10} = 0$

Now the algorithm

Let x be the input number.
While $x \neq 0$ do:
Save x mod 10 (remainder of the division by 10)
Integer divide of $x = \frac{x}{10}$ (scrap the fractional part)

Example:

Input: $x = 456$
Step 1: x mod $10 = 6$. Save 6 (units)
Step 2: $x = \frac{x}{10} = \frac{456}{10} = 45$
Is $x > 0$? Yes. Continue.
Step 3: x mod $10 = 5$. Save 5 (tens)
Step 4: $x = \frac{x}{10} = \frac{45}{10} = 4$
Is $x > 0$? Yes. Continue.
Step 5: x mod $10 = 4$. Save 4 (hundreds)
Step 6: $x = \frac{x}{10} = \frac{4}{10} = 0$
Is $x > 0$? No. Stop. We're done. Read it from last to first.

Decimal to Binary

Algorithm

Let x_{10} be the input number in base 10.
While $x \neq 0$ do:
  Save x mod 2 (remainder of the division by 2)
  Integer divide of $x = \frac{x}{2}$ (scrap the fractional part)
Remainders from last found to first make up the number x in base 2.

Example

Convert 11_{10} to binary

Input: $x = 11$

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$
Is $x > 0$? Yes. Continue.

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$
Is $x > 0$? Yes. Continue.
Step 3: $x \bmod 2 = 1$. Save 1

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Is $x > 0$? Yes. Continue.

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$
Is $x > 0$? Yes. Continue.
Step 3: $x \bmod 2 = 1$. Save 1
Step 4: $x = \frac{x}{2} = 2$
Is $x > 0$? Yes. Continue.
Step 5: $x \bmod 2 = 0$. Save 0

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Is $x > 0$? Yes. Continue.

Step 5: $x \bmod 2 = 0$. Save 0

Step 6: $x = \frac{x}{2} = 1$

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Is $x > 0$? Yes. Continue.

Step 5: $x \bmod 2 = 0$. Save 0

Step 6: $x = \frac{x}{2} = 1$

Is $x > 0$? Yes. Continue.

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$
Is $x > 0$? Yes. Continue.
Step 3: $x \bmod 2 = 1$. Save 1
Step 4: $x = \frac{x}{2} = 2$
Is $x > 0$? Yes. Continue.
Step 5: $x \bmod 2 = 0$. Save 0
Step 6: $x = \frac{x}{2} = 1$
Is $x > 0$? Yes. Continue.
Step 7: $x \bmod 2 = 1$. Save 1

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Is $x > 0$? Yes. Continue.

Step 5: $x \bmod 2 = 0$. Save 0

Step 6: $x = \frac{x}{2} = 1$

Is $x > 0$? Yes. Continue.

Step 7: $x \bmod 2 = 1$. Save 1

Step 8: $x = \frac{x}{2} = 0$

Example

Convert 11_{10} to binary

Input: $x = 11$
Step 1: $x \bmod 2 = 1$. Save 1
Step 2: $x = \frac{x}{2} = 5$
Is $x > 0$? Yes. Continue.
Step 3: $x \bmod 2 = 1$. Save 1
Step 4: $x = \frac{x}{2} = 2$
Is $x > 0$? Yes. Continue.
Step 5: $x \bmod 2 = 0$. Save 0
Step 6: $x = \frac{x}{2} = 1$
Is $x > 0$? Yes. Continue.
Step 7: $x \bmod 2 = 1$. Save 1
Step 8: $x = \frac{x}{2} = 0$
Is $x > 0$? No. Stop. We're done.

Example

Convert 11_{10} to binary

Input: $x = 11$

Step 1: $x \bmod 2 = 1$. Save 1

Step 2: $x = \frac{x}{2} = 5$

Is $x > 0$? Yes. Continue.

Step 3: $x \bmod 2 = 1$. Save 1

Step 4: $x = \frac{x}{2} = 2$

Is $x > 0$? Yes. Continue.

Step 5: $x \bmod 2 = 0$. Save 0

Step 6: $x = \frac{x}{2} = 1$

Is $x > 0$? Yes. Continue.

Step 7: $x \bmod 2 = 1$. Save 1

Step 8: $x = \frac{x}{2} = 0$

Is $x > 0$? No. Stop. We're done.

The number 11_{10} is 1011_2.

What we learned

- Number bases, we looked at decimal (base 10) and binary (base 2), there are other commonly used bases: octal (base 10), hexadecimal (16). Same rules and algorithms apply.
- Conversion from decimal to binary and binary to decimal.
- Next: signed integers and fractional (floating point) numbers.

Numerical data types

Whole numbers

We looked at whole, unsigned (positive) numbers. The basic storage unit used by modern computers is the bit (stands for binary digit). Numeric data types are characterized by:

- Capability of representing signed numbers (positive or negative) or unsigned (positive only).
- The number of bits used to represent numbers.
- Whole or fractional.

MATLAB numeric data types (integers)

- uint8 - unsigned integer, 8 bits. This is a byte. How many numbers can this type represent and what are they?

MATLAB numeric data types (integers)

- uint8 - unsigned integer, 8 bits. This is a byte. How many numbers can this type represent and what are they?
- uint16, uint32, uint64 - unsigned integers.
- int8, int16, int32, int64 - signed integers.

MATLAB numeric data types (floating point)

Fractions?

When all we have is a fixed number of bits (say, 64 bits) with possible values $\{0, 1\}$, how do we represent fractions?

MATLAB numeric data types (floating point)

Fractions?

When all we have is a fixed number of bits (say, 64 bits) with possible values $\{0, 1\}$, how do we represent fractions?

Basic idea

Use whole numbers to represent fractions? Remember $\mathbb{Q}$? In other words, we could represent a number in $\mathbb{Q}$ as $\frac{a}{b}$ where $a, b \in \mathbb{Z}$ and $b \neq 0$.
Option 1: we could set aside the first 32 bits for a and the second 32 bits for b if we had 64 bits to work with. Any problems with that approach?

MATLAB numeric data types (floating point)

Fractions?

When all we have is a fixed number of bits (say, 64 bits) with possible values $\{0, 1\}$, how do we represent fractions?

Basic idea

Use whole numbers to represent fractions? Remember $\mathbb{Q}$? In other words, we could represent a number in $\mathbb{Q}$ as $\frac{a}{b}$ where $a, b \in \mathbb{Z}$ and $b \neq 0$.
Option 1: we could set aside the first 32 bits for a and the second 32 bits for b if we had 64 bits to work with. Any problems with that approach?
Option 2: Scientific notation.

Floating point number representation

Floating point representation

$$number = mantissa * base^{exponent}$$

Here, $\{mantissa, base, exponent\} \in \mathbb{Z}$.
This notation allows for a much wider range of numbers to be represented.

Floating point number representation

Floating point representation

$$number = mantissa * base^{exponent}$$

Here, $\{mantissa, base, exponent\} \in \mathbb{Z}$.
This notation allows for a much wider range of numbers to be represented.

Example

$$1.2345 = 12345 * 10^{-4}$$

In practice

IEEE Standard for Floating-Point Arithmetic (IEEE 754). More details at: http://en.wikipedia.org/wiki/IEEE_floating_point.

MATLAB floating point types

- single - "single" precision (32 bits)
- double - "double" precision (64 bits)

MATLAB assumes the "double" as a default type for numbers.

MATLAB “whos” command

```
>> x = 10

x =

    10

>> whos
  Name          Size                Bytes  Class     Attributes

  x             1x1                     8  double

>>
```

Formatting numeric output in MATLAB

- format command (see textbook or MATLAB help for a complete reference). Examples:
  - format short (default). Short fixed decimal format, with 4 digits after the decimal point. Example: 3.1416
  - format long. Long fixed decimal format, with 15 digits after the decimal point for double values, and 7 digits after the decimal point for single values. Example: 3.141592653589793
  - format longG. The more compact of long fixed decimal or scientific notation, with 15 digits for double values, and 7 digits for single values. Example: 3.14159265358979
  - format shortE. Short scientific notation, with 4 digits after the decimal point. Example: 3.1415e+00
  - format longE. Long scientific notation, with 15 digits after the decimal point for double values, and 7 digits after the decimal point for single values. Example: 3.141592653589793e+00