CS 100: Bits and Things

Chris Kauffman

Week 1-2

Logistics

Reading: Pattern on the Stone, Ch 0 and 1

HW 1 Posted

- Due next Thursday
- Will cover many needed skills today
- Where can you see the HW?

Goals

- Representing stuff with bits
- Boolean logic and bits
- Primitive pieces that comprise computers

Hot Seats

- Each session, first few rows are hot seats
- First come, first serve (adjust if needed)
- Just try: answer questions, give feedback
- Don't want/need participation, sit elsewhere
- Up to 3% overall bonus
  - Luke and Leia have 20 part pts, max in class, 3% bonus each
  - Han and Chewie have 10 part pts, 1.5% bonus each
  - Greedo has 0 part pts, 0.0% bonus
- Scoring described in Syllabus
- May be a few other opportunities for participation

Code.org

- Sign up for Code.org
  - Direct link: `http://studio.code.org/join/WFPGRG`
  - Course Code `WFPGRG`
- Use your GMU Email address and full name
- HW2 will be to finish the course

Bits

Bit: a spot to store a TRUE or FALSE value

- Usually written as 1 for TRUE and 0 for FALSE
- Real data (pics, movies, songs, text) require *many* bits
- A byte is 8 bits grouped together

Bit Index	0	1	2	3	4	5	6	7
Bit Value	0	1	1	0	0	0	0	1
Boolean	False	True	True	False	False	False	False	True

What does the bit sequence `01100001` mean? That depends...

Bits

Bits can represent anything but require interpretation
Example: the byte (8 bits) 01100001 could be...

- As an unsigned integer: $1 + 2^5 + 2^6 = 97$
- As an ASCII character: the letter a
- The amount of green in the color of a single pixel
- Various other things if you decide it does

Important

If you are given some bits, you will also need to know how to interpret them. We'll discuss some common interpretations.

Numbers

- Bits are frequently used for numbers
- Common style: integers numbers that are in base 2
- Base 2 numbering works like Base 10 does but only 0 and 1 shows up

Base 10 Example:

$$
\begin{aligned}
80,345 = & 5 \times 10^0 + & 5 \\
& 4 \times 10^1 + & 40 \\
& 3 \times 10^2 + & 300 \\
& 0 \times 10^3 + & 0 \\
& 8 \times 10^4 & 80,000
\end{aligned}
$$

Base 2 Example:

$$
\begin{aligned}
11001_2 = & 1 \times 2^0 + & 1 \\
& 0 \times 2^1 + & 0 \\
& 0 \times 2^2 + & 0 \\
& 1 \times 2^3 + & 8 \\
& 1 \times 2^4 + & 16 \\
= & 1 + 8 + 16 & = 25
\end{aligned}
$$

So, $11001_2 = 25_{10}$

Try

Base 2 Example:

$$
\begin{aligned}
11001 = & 1 \times 2^0 + & 1 \\
& 0 \times 2^1 + & 0 \\
& 0 \times 2^2 + & 0 \\
& 1 \times 2^3 + & 8 \\
& 1 \times 2^4 + & 16 \\
= & 1 + 8 + 16 & = 25
\end{aligned}
$$

So, $11001_2 = 25_{10}$

Try With a Pal

Convert the following two numbers from base 2 (binary) to base 10 (decimal)

- 111
- 11010
- 01100001

The Other Direction: Base 10 to Base 2

Converting a number from base 10 to base 2 is easily done using repeated division by 2; keep track of remainders

Convert 124 to base 2:

$$\begin{aligned} 124 \div 2 &= 62 && \text{rem } 0 \\ 62 \div 2 &= 31 && \text{rem } 0 \\ 31 \div 2 &= 15 && \text{rem } 1 \\ 15 \div 2 &= 7 && \text{rem } 1 \\ 7 \div 2 &= 3 && \text{rem } 1 \\ 3 \div 2 &= 1 && \text{rem } 1 \\ 1 \div 2 &= 0 && \text{rem } 1 \end{aligned}$$

- Last step got 0 so we're done.
- Binary digits are in remainders in reverse
- Answer: 1111100
- Check:
  $0 + 0 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 = 4 + 8 + 16 + 32 + 64 = 124$

Try

124_{10} to Base 2

$$
\begin{aligned}
124 \div 2 &= 62 && \text{rem } 0 \\
62 \div 2 &= 31 && \text{rem } 0 \\
31 \div 2 &= 15 && \text{rem } 1 \\
15 \div 2 &= 7 && \text{rem } 1 \\
7 \div 2 &= 3 && \text{rem } 1 \\
3 \div 2 &= 1 && \text{rem } 1 \\
1 \div 2 &= 0 && \text{rem } 1
\end{aligned}
$$

- Remainders in reverse
- $124_{10} = 1111100_2$

Conversion

Convert the 19_{10} from base 10 to base 2: should only have 1's and 0's.

Algorithm

Describe the series of steps to do conversion from base 10 to base 2. Write it as an algorithm

Decimal to Binary Algorithm

Here is one version of the algorithm

```
Input X, a decimal number

set ANSWER to be "" (empty)
repeat while X is not 0:
    divide X by 2 to get the QUOTIENT and REMANDER
    Set X to be QUOTIENT
    Prepend REMAINDER to the LEFT side of ANSWER

ANSWER now contains the binary representation for X
```

Additional info on binary and decimal conversions from the University of New Mexico.

Bits for Letters

- Bits don't have to be interpretted as numbers
- Could be letters instead
- Typically assign each letter a bit string
- How many letters in the English (Latin) alphabet?
- How many bits would we need to represent them?

English Characters

Scheme 1: Placement of 1

A	1	00000000000000000000000001
B	10	00000000000000000000000010
C	100	00000000000000000000000100
D	1000	00000000000000000000001000
E	10000	00000000000000000000010000
..	...	
Z	10000000000000000000000000	10000000000000000000000000

26 bits for 26 Characters

- Okay, but seems like a lot of 0's...

English Characters

Scheme 2: Each letter is also a number

A	0	0	00000
B	1	1	00001
C	2	10	00010
D	3	11	00011
E	4	100	00100
F	5	101	00101
..		...	
Z	26	11001	11001

5 bits for 26 characters

- 5 bits could handle up to 32 characters

In general: X things can be represented by N bits where $X \leq 2^N$

For Next Time

- "Pattern": Ch 1-3
- "Think": Play with turtle graphics
- Start working on HW 1