Number Representations and the Division Algorithm

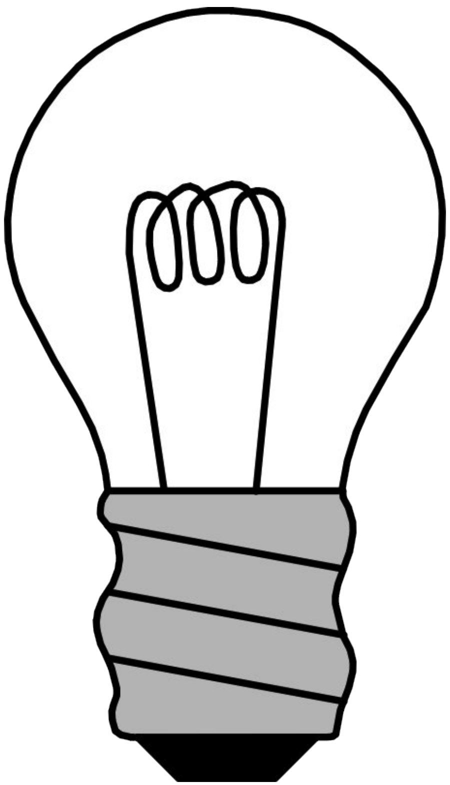
CS 2800: Discrete Structures, Spring 2015

Sid Chaudhuri

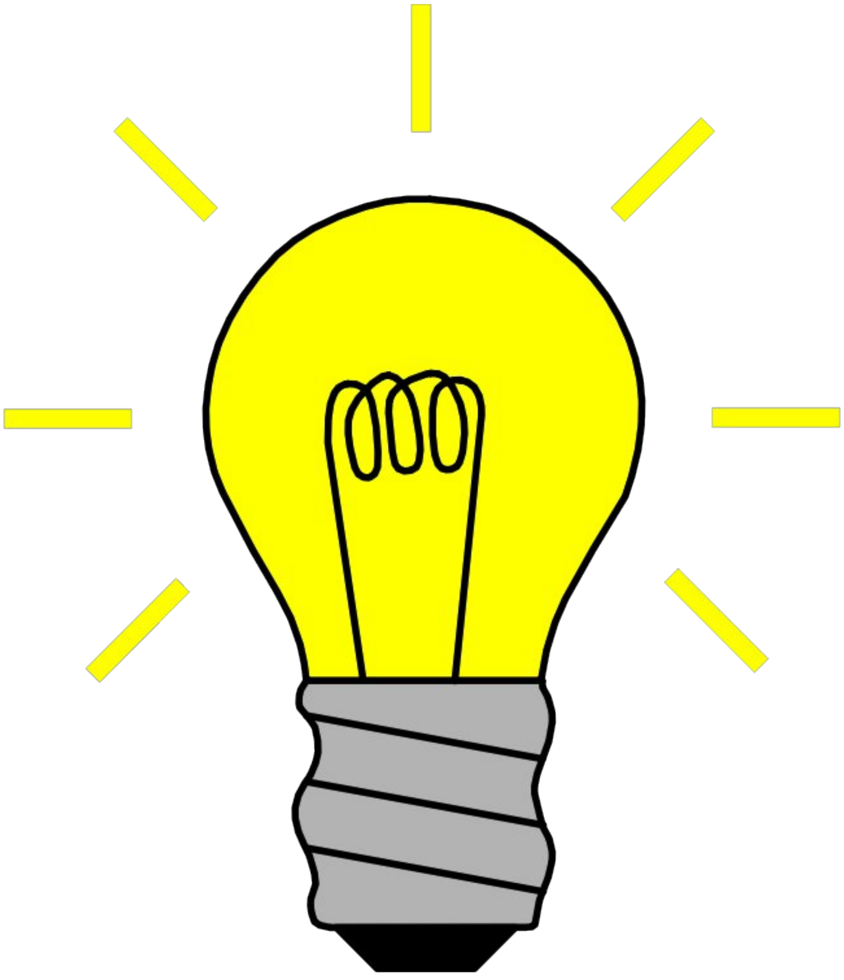
There are only 10 types of people in the world.

Those who understand binary, and those who don't

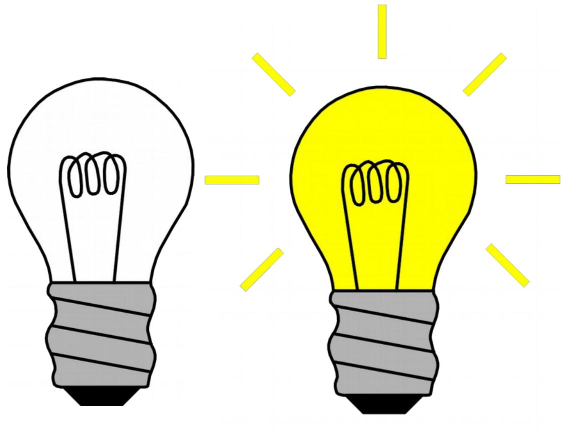
Binary Digits

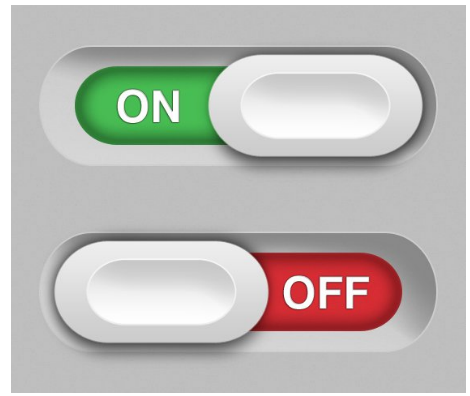


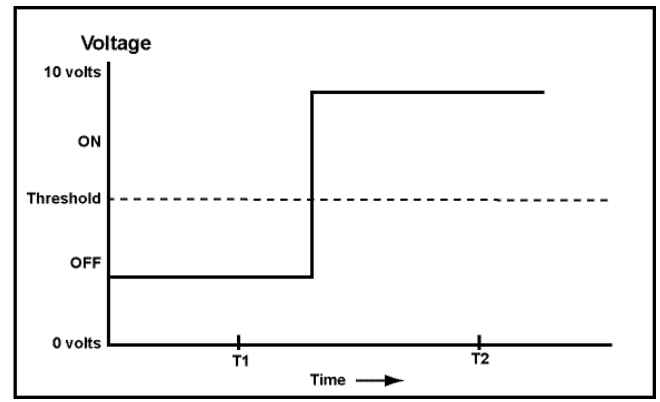
Binary Digits



Binary Digits







True
False

0
1

?

openclipart.org, imgbuddy.com, chortle.ccsu.edu, speechdudes.wordpress.com, ncetm.org.uk

Binary representations of numbers

$$1\ 0\ 1\ 0\ 1_2$$

$$1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$

$$= 21_{10}$$

Numbers in base b

Each a_i is a digit between 0 and $b-1$

$$\overbrace{a_4 \; a_3 \; a_2 \; a_1 \; a_0}$$

$$a_4 \times b^4 + a_3 \times b^3 + a_2 \times b^2 + a_1 \times b^1 + a_0 \times b^0$$

Common bases:

Binary (2), **Ternary** (3), **Octal** (8), **Decimal** (10), **Hexadecimal** (16)

All rules of arithmetic remain exactly the same,
just remember 10_b is b

Common bases

- **Binary** (base 2)
  - Digits: 0, 1
- **Ternary** (base 3)
  - Digits: 0, 1, 2
- **Octal** (base 8)
  - Digits: 0, 1, 2, 3, 4, 5, 6, 7
- **Decimal** (base 10)
  - Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- **Hexadecimal** (base 16)
  - Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (= 10_{10}), B (=11_{10}), C (=12_{10}), D (=13_{10}), E (=14_{10}), F (=15_{10})

Conversions to/from decimal

- Converting from base b to decimal
  - Add up the powers of b as in the previous slide
- Converting from decimal to base b
  - Divide by b and write down the remainder
  - Repeat with the quotient, writing down the remainders ***right*** to ***left***

Example conversions worked out on the board

“Division Algorithm” (not really an algorithm)

- Theorem: Given any integer a, and a positive integer b, there exist integers q (the “quotient”), and r (the “remainder”), such that
  - $0 \leq r < b$, and
  - $a = qb + r$

- Proof: By induction!

we'll prove it only for non-negative a –
the proof for negative a is similar

Proof of Division Algorithm

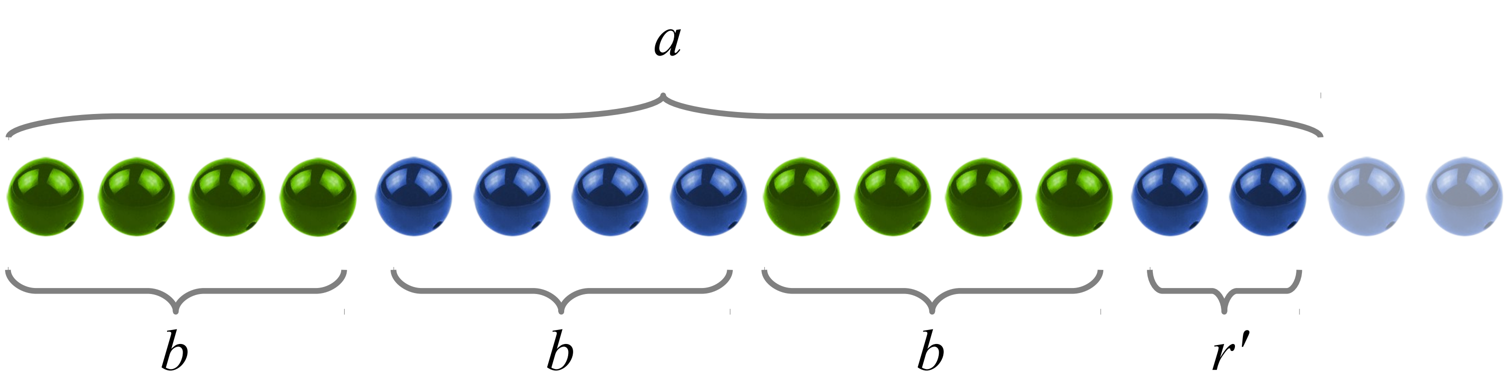
(for non-negative a)

- We will do induction on a
- $S(a)$ = "for the given a, and any b, the theorem is true"
- Base case:
  - When $a = 0$, choose $q = 0$, $r = 0$
  - Clearly $0 \leq r < b$ (since $b > 0$) and $a = qb + r$
- Inductive hypothesis: Given a, we have $a = q'b + r'$ for q' and r' satisfying the conditions

Proof of Division Algorithm

(for non-negative a)

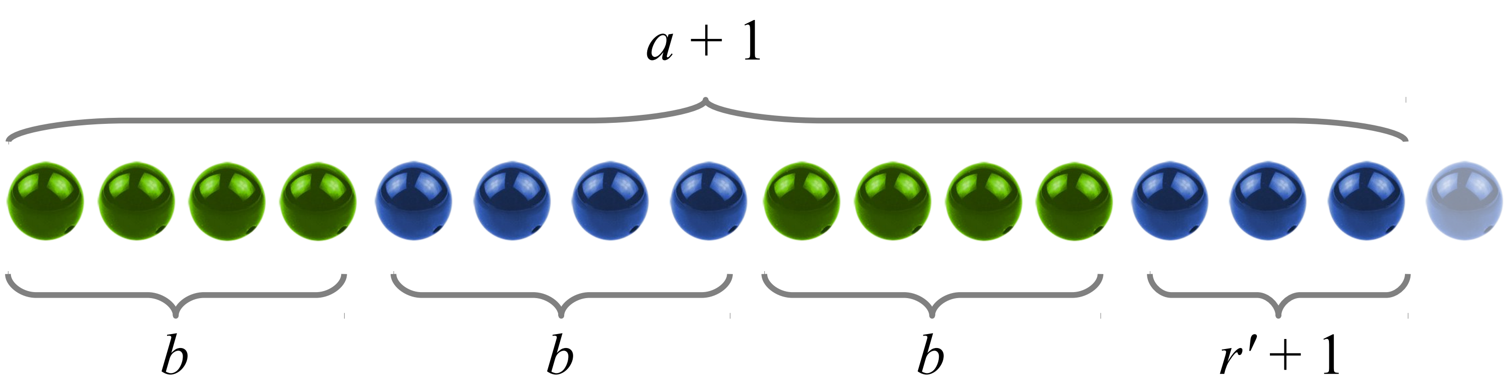
- Inductive step: Two cases
  - Case 1: $r' < b - 1$



Proof of Division Algorithm

(for non-negative a)

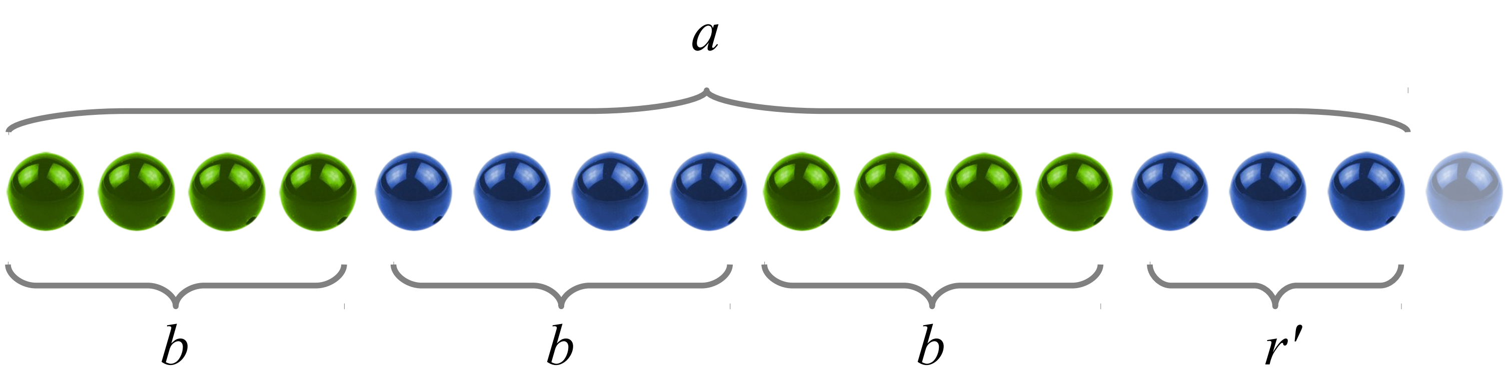
- Inductive step: Two cases
  - Case 1: $r' < b - 1$
    - Choose $q = q', r = r' + 1$
    - Clearly $0 \leq r < b$
    - ... and $a + 1 = q'b + r' + 1 = q'b + (r' + 1) = qb + r$



Proof of Division Algorithm

(for non-negative a)

- Inductive step: Two cases
  - Case 2: $r' = b - 1$

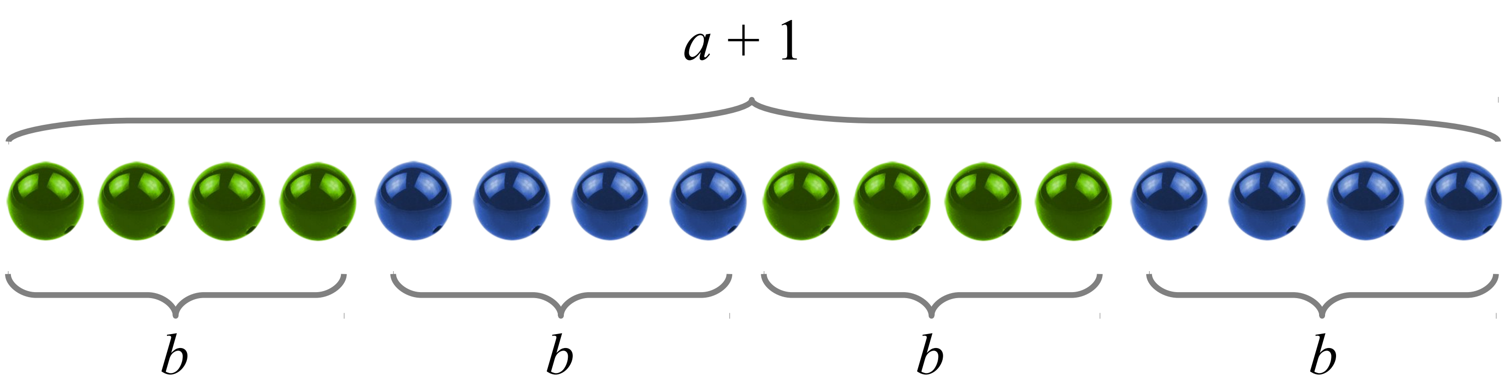


Proof of Division Algorithm

(for non-negative a)

- Inductive step: Two cases
  - Case 2: $r' = b - 1$
    - Choose $q = q' + 1, r = 0$
    - Clearly $0 \leq r < b$
    - … and $a + 1 = q'b + r' + 1 = q'b + (b - 1) + 1 = (q' + 1)b + 0$
      $= qb + r$

Hence proved by induction!



Thought for the Day #1

Write out the proof for negative a

Quotient and Remainder are Unique

- Proof: Assume $a = qb + r = q'b + r'$
  - Then $(q - q')b = r' - r$
  - Since r and r' are between 0 and $b - 1$, we have

    $-b < (r' - r) < b$
  - Hence $-b < (q - q')b < b$
  - Since $b > 0$, we can divide to get

    $-1 < (q - q') < 1$
  - Hence $q = q'$ (since $q - q'$ is an integer)
  - ... and $r = r'$ (since $r' - r = (q - q')b = 0$)
- Implies that the representation of a number in a given base is also unique! (Prove!)