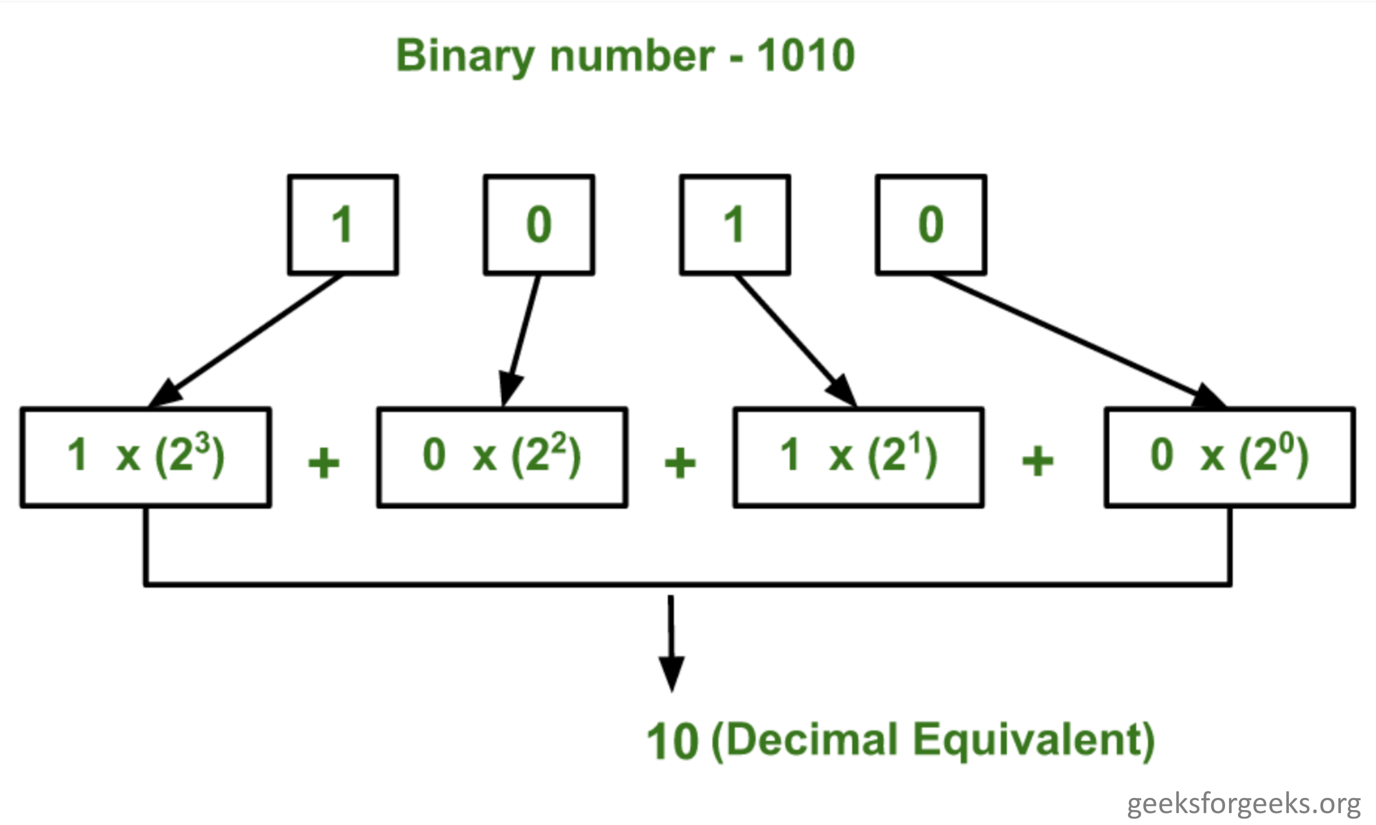
1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0

CSCI 2824: Discrete Structures

Lecture 1: Intro & Binary Representation of Numbers

Rachel Cox

Department of Computer Science



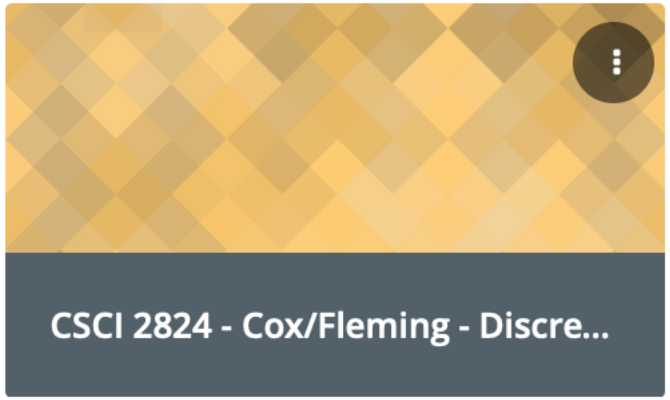
1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0

Course Logistics – Platforms

1) Moodle – Online Homework, Online Quizlets, Grades

Enrollment Key:

CSCI 2824-FALL 2019



2) Piazza– Class discussion forum

CSCI 2824 ▾ Q & A Resources Statistics Manage Class

University of Colorado at Boulder - Fall 2019

CSCI 2824: Discrete Structures

3) Gradescope – Submission of written homework

gradescope

Course Logistics – Platforms

Moodle – Online Homework, Online Quizlets, Grades

moodle.cs.colorado.edu

Enrollment key (case sensitive):

section 001 – 9am:
section 002 – 11am: CSCI2824-FALL2019

Course Logistics – Platforms

Course Webpage – Piazza

https://piazza.com/colorado/fall2019/csci2824/home

- Office Hours
- General Info
- Instead of emailing, post questions to Piazza – it'll be faster!
- Announcements
- Homework & Solutions

Course Logistics – Platforms

Gradescope – I will enroll you. You’ll receive an email once I’ve enrolled you. This is where you will turn in your written homework.

Course Logistics – Grading

Weekly Homework (30%)

- Half Written, Half Online

Quizlets (10%)

- Online

Two Midterms (20% each)

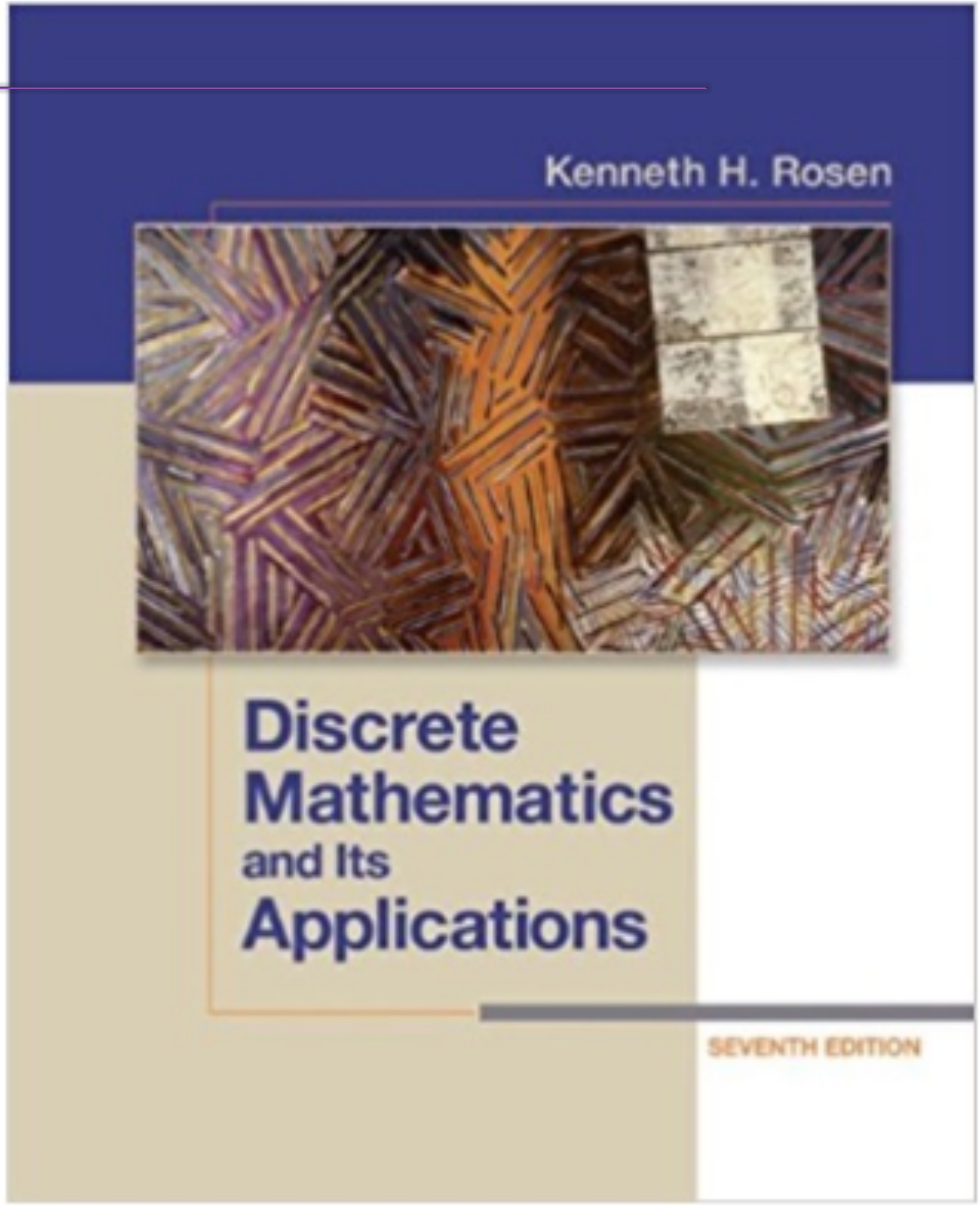
- October 1st , November 5th

Final Exam (20%)

- Section 001 – 9am: Wednesday December 18th , 1:30-4:00pm
- Section 002 – 11am: Sunday December 15th , 1:30-4:00pm

Course Logistics – Book

<u>Textbook</u> – *Discrete Mathematics and Its Applications*, 7th Ed. by Kenneth H. Rosen



101100001000 101100001000 101100001000 101100001000 101100001000

What is Discrete Structures?

Discrete

- Logic
- Combinatorics
- Discrete Probability
- Recursion
- Sets
- Sequences
- Graph Theory

.
.
.

NOT Discrete

- Derivatives
- Integrals
- … things that involve infinitesimals

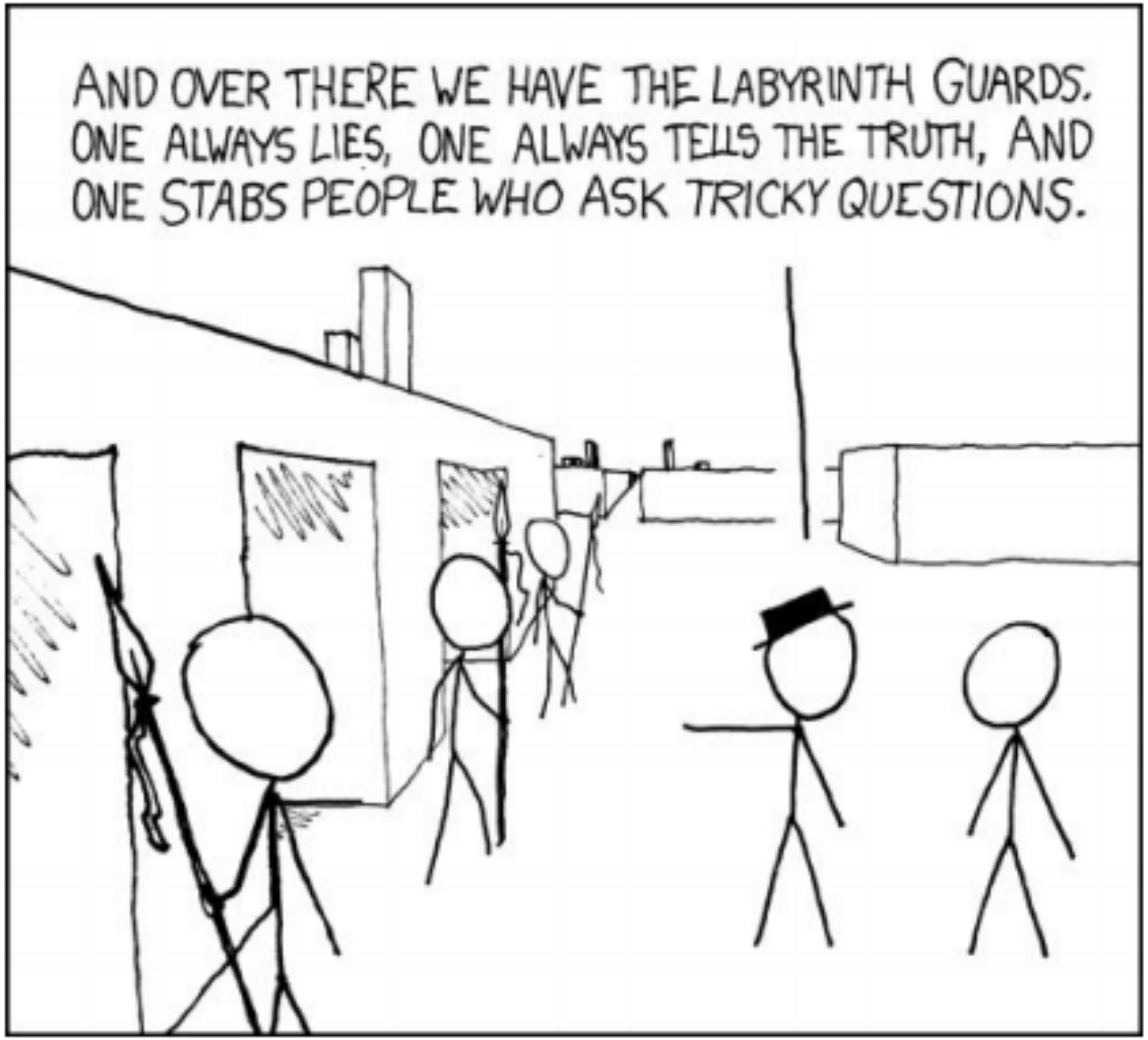
101100001000 101100001000 101100001000 101100001000 101100001000

Logic:

“Two doors” riddle:

- Two doors, guarded by two guards.
- One door goes where you want to, but the other leads to certain death.
- One always lies, and one always tells the truth.

- **How can you ask**
  <u>only one of them</u>
  <u>only one question</u>
  to discover which door is which?



Discrete Probability:

e.g. The Monte Hall Problem

Three doors problem

- There are three doors.
- One has a nice prize behind it…
- … and the other two have goats.
- You get to pick a door and will be awarded the prize behind it.
- Then the host reveals a goat behind one of the other two doors.
- You now have the option to stick with your original door or switch.

- **Should you stick with your original door or switch? Or does it not matter?**

Recursion:

e.g. The Tower of Hanoi

- Recursive solutions can be elegant and lead to more readable code
- Recursion may be frowned upon due to memory stack issues
- However, developing a recursive solution and then translating it to an iterative solution (loops) may be very helpful

101100001000 101100001000 101100001000 101100001000 101100001000

Binary Representation of Numbers

0 1

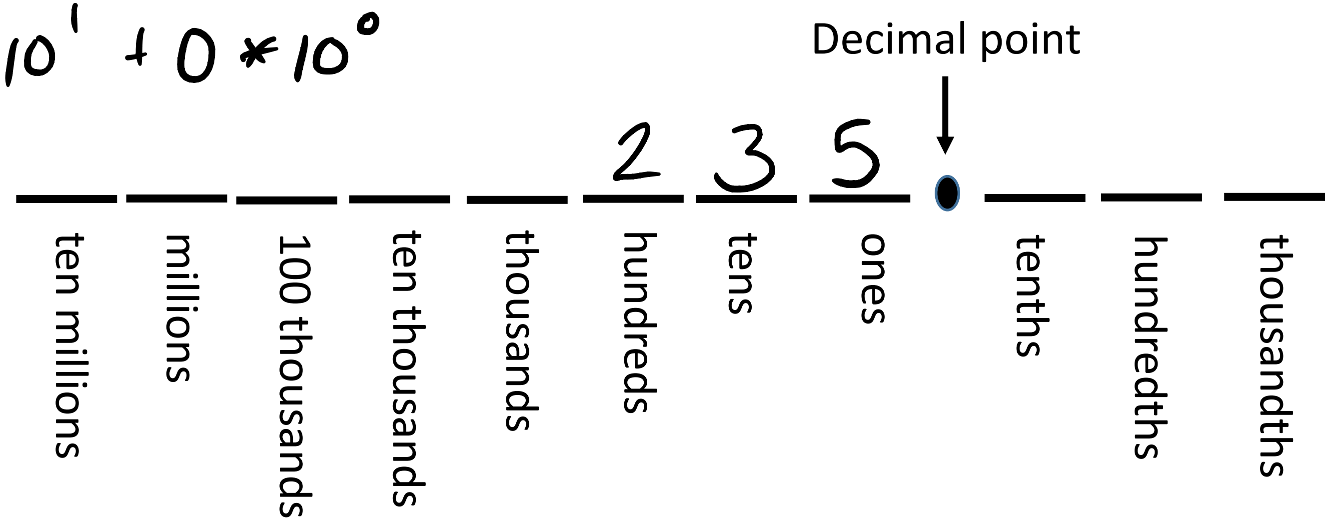
101100001000 101100001000 101100001000 101100001000 101100001000

What's in a Number?

<u>Example</u>: Consider the numbers **235** and **1130**. Assume they are both "decimal".
How can we expand them as powers of 10?

$$235 = 200 + 30 + 5$$
$$= 2*100 + 3*10 + 5*1$$
$$= 2*10^2 + 3*10^1 + 5*10^0$$

$$1130 = 1*10^3 + 1*10^2 + 3*10^1 + 0*10^0$$



0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Binary Number System

- Representing numbers in “base 2”
- Denote a binary number by $(binary\ number)_2$

Radix point

128s	sixty-fours	thirty-twos	sixteens	eights	fours	twos	ones	.	halves	quarters	eighths

Decimal point

ten millions	millions	100 thousands	ten thousands	thousands	hundreds	tens	ones	.	tenths	hundredths	thousandths

Powers of 2:

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$
$$2^4 = 16$$
$$2^5 = 32$$
$$2^6 = 64$$
$$2^7 = 128$$
$$2^8 = 256$$

Powers of 10:

$$10^0 = 1$$
$$10^1 = 10$$
$$10^2 = 100$$
$$10^3 = 1000$$
$$10^4 = 10{,}000$$
$$10^5 = 100{,}000$$
$$10^6 = 1{,}000{,}000$$
$$10^7 = 10{,}000{,}000$$
$$10^8 = 100{,}000{,}000$$

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Example: Convert 235 from decimal to binary

Powers of 2:

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$

0	0	0	0	0	0	0
0	0	0	0	1		0
0	0	0	1	1		0
0	1	0	0			0
1	1	1	1			0
0	0	1				1
0		1				0
0						0
						1

$$
\begin{aligned}
235 &= 128 + 107 \\
&= 128 + 64 + 43 \\
&= 128 + 64 + 32 + 11 \\
&= 128 + 64 + 32 + 8 + 3 \\
&= 128 + 64 + 32 + 8 + 2 + 1 \\
&= 2^7 + 2^6 + 2^5 + 2^3 + 2^1 + 2^0 \\
&= \underline{1} * 2^7 + \underline{1} * 2^6 + \underline{1} * 2^5 + \underline{0} * 2^4 + \underline{1} * 2^3 + \underline{0} * 2^2 + \underline{1} * 2^1 + \underline{1} * 2^0
\end{aligned}
$$

1	1	1	0	1	0	1	1	.			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
				eights	fours	twos	ones				

0	0	0	0	0	0	0
0	0	0	0	1		0
0	0	0	1	1		0
0	1	0	0			0
1	1	1	1			0
0	0	1				1
0		1				0
0						0
						1

<u>Example:</u> Convert 235 from decimal to binary in a more systematic way.

$$235 = 1 + 234$$

$$235 = 1 + 2 \cdot 117$$

$$= 1 + 2(1 + 116)$$

$$= 1 + 2 + 2 \cdot 116$$

$$= 1 + 2 + 2^2 \cdot 58$$

$$= 1 + 2 + 2^3 \cdot 29$$

$$= 1 + 2 + 2^3(1 + 28)$$

$$= 1 + 2 + 2^3 + 2^3 \cdot 28$$

$$\frac{235 - 1}{2} = 117$$

Number Theory

Factoring

Algorithm

				1	0	1	1	.			
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

An Algorithm for Converting Decimal Integers to Binary

Let N be a nonnegative integer. Move from right to left.

Is N even? or odd?

If N is even, set bit to 0, reset $N = \frac{N}{2}$

If N is odd, set bit to 1, reset $N = \frac{N-1}{2}$

Move left to the next bit

Repeat until N = 0

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

0	0	0	0	0	0	0
0	0	0	0	1		0
0	0	0	1	1		0
0	1	0	0			0
1	1	1	1			0
0	0	1				1
0		1				0
0						0
						1

<u>Example:</u> Convert 1130 from decimal to binary using the algorithm we just defined.

• Radix point

N			
1130	even	set 2^0 bit = 0	$N = \frac{1130}{2} = 565$
565	odd	set 2^1 bit = 1	$N = \frac{565-1}{2} = 282$
282	even	set 2^2 bit = 0	$N = 141$
141	odd	set 2^3 bit = 1	$N = 70$
70	even	set 2^4 bit = 0	$N = 35$
35	odd	set 2^5 bit = 1	$N = 17$
17	odd	set 2^6 bit = 1	$N = 8$
8	even	set 2^7 bit = 0	$N = 4$
4	even	set 2^8 bit = 0	$N = 2$
2	even	set 2^9 bit = 0	$N = 1$
1	odd	set 2^{10} bit = 1	$N = \frac{1-1}{2} = 0$

e.g. $00347 = 347$

• you can truncate the leading zeros

$$(1130)_{10} = (10001101010.)_2$$

0	0	0	0	0	0	0
0	0	0	0	1		0
0	0	0	1	1		0
0	1	0	0			0
1	1	1	1			0
0	0	1				1
0		1				0
0						0
						1

<u>Example:</u> Convert 160 and 161 from decimal to binary using the algorithm we just defined.

N	
160	0
80	0
40	0
20	0
10	0
5	1
2	0
1	1

$$(160)_{10} = (10100000)_2$$

$$(161)_{10} = (10100001)_2$$

0	0	0	0	0	0	0
0	0	0	0	1		0
0	0	0	1	1		0
0	1	0	0			0
1	1	1	1			0
0	0	1				1
0		1				0
0						0
						1

<u>Example:</u> Convert 1100101 from binary to decimal.

$$\frac{1}{2^6}\ \frac{1}{2^5}\ \frac{0}{2^4}\ \frac{0}{2^3}\ \frac{1}{2^2}\ \frac{0}{2^1}\ \frac{1}{2^0}$$

$$(1100101)_2 = 64 + 32 + 4 + 1 = (101)_{10}$$

<u>Example:</u> What’s the largest number you can store as a 32-bit signed int?

Marin Mersenne

Born	8 September 1588 Oizé, Maine
Died	1 September 1648 (aged 59) Paris
Nationality	French
Known for	Acoustics, Mersenne primes