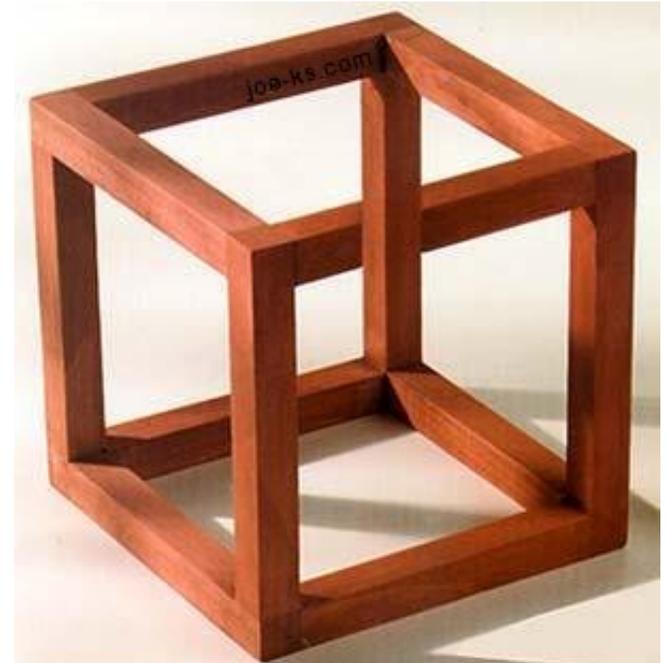
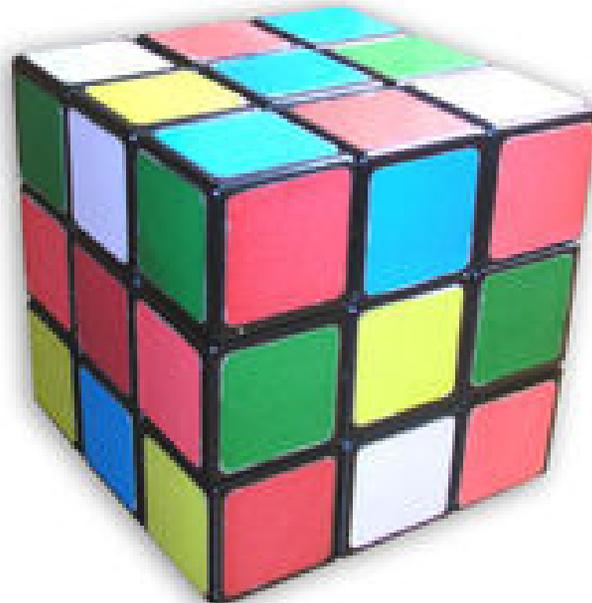


# Easy, Hard, Impossible!

A Lecture in CE Freshman Seminar Series:  
Ten Puzzling Problems in Computer Engineering

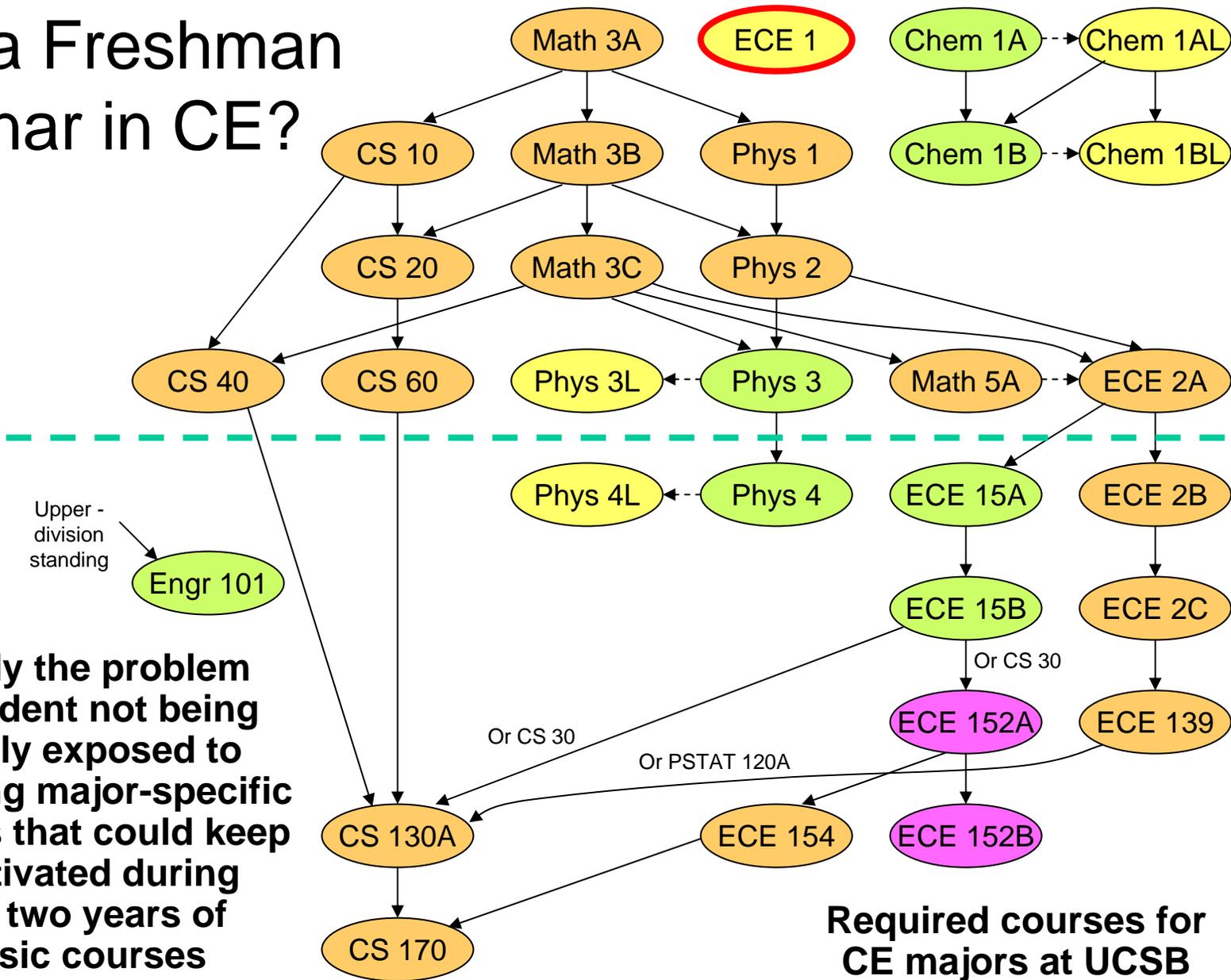
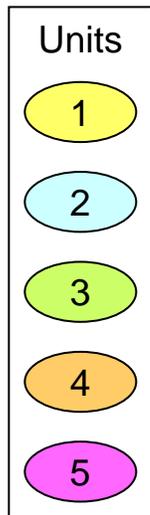


# About This Presentation

This presentation belongs to the lecture series entitled “Ten Puzzling Problems in Computer Engineering,” devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

<b>Edition</b>	<b>Released</b>	<b>Revised</b>	<b>Revised</b>
<b>First</b>	Apr. 2007		

# Why a Freshman Seminar in CE?

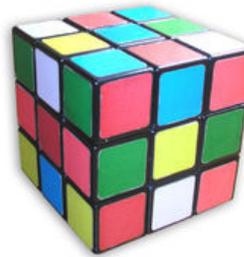


To remedy the problem of CE student not being sufficiently exposed to interesting major-specific problems that could keep them motivated during their first two years of taking basic courses

# What Are Puzzling Problems?

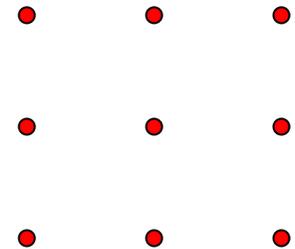
## A puzzling problem:

- ☞ looks deceptively simple, but ...
- ☞ appears very difficult, or even impossible, but is readily tamed with the correct insight



Rotate faces until each face is single-colored

Connect all dots using four straight lines, without lifting your pen



Many engineering problems are puzzle-like (especially in CE)

Because of a long-standing interest in mathematical puzzles, I designed this course that combines my personal and professional passions

Each lecture starts with one or more puzzles

We will try to solve the puzzles and discuss possible solution methods

I introduce you to CE problems that are related to the puzzles

# Course Expectations and Resources

**Grading:** Pass/Fail, by attendance and class participation

0 or 1 absence: Automatic “Pass”

2 absences: “Pass” if you had prior approval for your second absence or else had strong participation in class

3 absences: Can earn a “Pass” by taking a final oral exam covering the missed lectures

4 or more absences: “Fail”

Attendance will be taken at the beginning of each class period. If you arrive late, it is your responsibility to see the instructor after class.

Course website: [http://www.ece.ucsb.edu/Faculty/Parhami/ece\\_001.htm](http://www.ece.ucsb.edu/Faculty/Parhami/ece_001.htm)  
(PowerPoint and pdf presentations, addresses of relevant websites)

Instructor’s office hours: T 2:00-3:30, R 10:00-11:30, Room 5155 HFH

Now, on to  
Today's  
Topic of  
Discussion:  
Easy,  
Hard,  
Impossible



"It's never easy. The trick is to make it look easy."

# Easy: Euclid Sequences

Form a sequence of number pairs (integers) as follows:

Begin with any two positive numbers as the first pair

In each step, the next number pair consists of

(1) the smaller of the current pair of values, and (2) their difference

Stop when the two numbers in the pair become equal

(10, 15) (10, 5) (5, 5)

(22, 6) (6, 16) (6, 10) (6, 4) (4, 2) (2, 2)

(9, 23) (9, 14) (9, 5) (5, 4) (4, 1) (1, 3) (1, 2) (1, 1)

Why is the process outlined above guaranteed to end?

**Challenge:** Repeat this process for a few more starting number pairs and see if you can discover something about how the final number pair is related to the starting values

# Not So Easy: Fibonacci Sequences

Form a sequence of numbers (integers) as follows:  
Begin with any two numbers as the first two elements  
In each step, the next number  
    is the sum of the last two numbers already in the sequence  
Stop when you have generated the  $j$ th number ( $j$  is given)

5   16   21   37  $j = 4$

2   0   2   2   4   6   10   16   26  $j = 9$

1   1   2   3   5   8   13   21   34   55   89   144  $j = 12$

**Challenge:** See if you can find a formula that yields the  $j$ th number directly (i.e., without following the sequence) when we begin with 1 1

# Very Hard: Collatz Sequences

Form a sequence of numbers (integers) as follows:

Begin with a given number

To find the next number in each step,

halve the current number if it is even or triple it and add 1 if it is odd

5 16 8 4 2 1

The pattern 4 2 1 repeats  
(5 steps to reach the end)

22 11 34 17 52 26 13 40 20 10 5 ... 1 (15 steps)

9 28 14 7 22 11 ... 1 (19 steps)

**Challenge:** Repeat this process for 27 and some other starting values. See if you can discover something about how various sequences end; i.e., do all sequences end in the same way, fall into several categories, or do not show any overall pattern at all?

Reference: [http://en.wikipedia.org/wiki/Collatz\\_conjecture](http://en.wikipedia.org/wiki/Collatz_conjecture)

# Easy, Not So Easy, Very Hard

**Euclid:** Form a sequence of number pairs (integers) as follows:  
Begin with any two positive numbers as the first pair  
In each step, the next number pair consists of  
    (1) the smaller of the current pair of values, and (2) their difference  
Stop when the two numbers in the pair become equal

**Fibonacci:** Form a sequence of numbers (integers) as follows:  
Begin with any two numbers as the first two elements  
In each step, the next number  
    is the sum of the last two numbers already in the sequence  
Stop when you have generated the  $j$ th number ( $j$  is given)

**Collatz:** Form a sequence of numbers (integers) as follows:  
Begin with a given number  
To find the next number in each step,  
    halve the current number if it is even or triple it and add 1 if it is odd

# What Makes a Computational Problem Easy?

**Euclid:** Form a sequence of number pairs (integers) as follows:  
Begin with any two positive numbers as the first pair  
In each step, the next number pair consists of  
    (1) the smaller of the current pair of values, and (2) their difference  
Stop when the two numbers in the pair become equal

**Algorithm:**  $\gcd(x, y)$ , greatest common divisor of  $x$  and  $y$

If  $x = y$ , then output  $x$  and stop

Otherwise, compute  $\gcd(\min(x, y), |x - y|)$

**Analysis:** Number of steps in the worst case

$(1, n)$      $(1, n - 1)$      $(1, n - 2)$     . . .     $(1, 1)$      $(n - 1 \text{ steps})$

When the number of steps is a *polynomial function* of the problem size, the problem is considered computationally easy or *tractable*

The idea is that modern computers can perform many billions of computations per second, so even  $n^2$  or  $n^5$  steps may be manageable

# Easy Problems: Degrees of Ease

Consider the computation of  $x \bmod y$  (remainder of dividing  $x$  by  $y$ ), where  $x$  and  $y$  are positive integers

**Algorithm:**  $rem(x, y)$ , remainder of dividing  $x$  by  $y$

If  $x < y$ , then output  $x$  and stop

Otherwise, compute  $rem(x - y, y)$

**Analysis:** Number of steps in the worst case

$(n, 1) \quad (n - 1, 1) \quad (n - 2, 1) \quad . . . \quad (0, 1)$

$(n \text{ steps})$

For example, if  $n \approx 10^{16}$  (i.e., a 16-digit decimal number), and if each computation step takes 1 ns ( $10^{-9}$  s), then the execution time of this algorithm would be  $10^{16}/10^9 \text{ s} = 10^7 \text{ s} \approx 2778 \text{ hr} \approx 116 \text{ days} \approx 4 \text{ mo}$

Long division would yield the answer much quicker, in time that depends on the number of digits in  $n$  (i.e.,  $\log n$ ) and not on  $n$  itself

Computer scientists and engineers pursue more efficient algorithms

# Some Difficult-Looking Problems Are Easy

**Fibonacci:** Form a sequence of numbers (integers) as follows:  
Begin with any two numbers as the first two elements  
In each step, the next number  
    is the sum of the last two numbers already in the sequence  
Stop when you have generated the  $j$ th number ( $j$  is given)

**Algorithm:**  $Fib(j)$ , the  $j$ th Fibonacci number

Set  $x := 1$  and  $y := 1$

Repeat  $j - 2$  times: Set  $z = x + y$ ;  $x := y$ ;  $y := z$

( $j - 2$  steps)

**Quicker solution for large  $j$**

The golden ratio  $\left[ \frac{1 + \sqrt{5}}{2} \right] = 1.61803\dots$

$$Fib(j) = \frac{\left[ \frac{1 + \sqrt{5}}{2} \right]^j - \left[ \frac{1 - \sqrt{5}}{2} \right]^j}{\sqrt{5}}$$

In the field of computational complexity, degrees of ease or difficulty are measured with reference to the best known algorithm for the problem

# Some Easy-Looking Problems Are Difficult

**Collatz:** Form a sequence of numbers (integers) as follows:  
Begin with a given number  
To find the next number in each step,  
    halve the current number if it is even or triple it and add 1 if it is odd

**Algorithm:**  $\text{Collatz}(n)$ , Collatz sequence for  $n$

Print  $n$  and set  $x := n$

While  $x > 1$  repeat

    If  $x$  is even then  $x := x/2$  else  $x := 3x + 1$

    In either case, print the new value of  $x$

Number of steps,  
or even whether  
it is always finite,  
is unknown

**Collatz's conjecture:** For any starting value, the sequence reaches 1

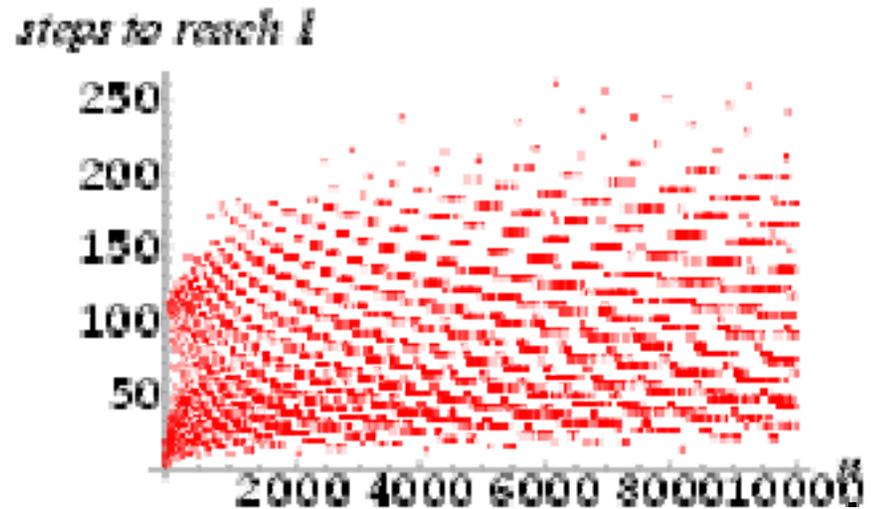
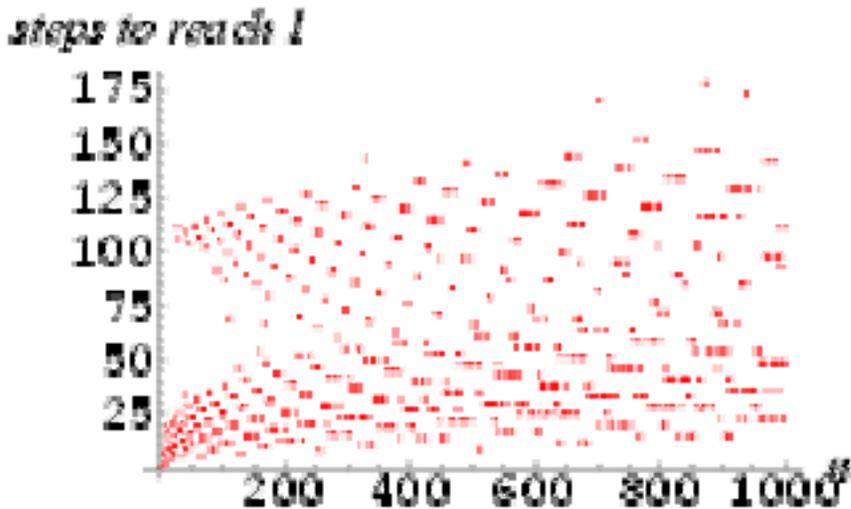
The truth of this conjecture has not yet been verified or contradicted

Evidence suggesting truth: experimentally verified for  $n$  up to  $2.7 \times 10^{16}$ ;  
next odd number in sequence is on average about  $\frac{3}{4}$  of previous one

# Getting a Handle on Difficult Problems

Use the computer to experiment and find solutions for many instances; then, try to generalize from the results and patterns observed

Number of steps for the Collatz sequence to reach 1, as a function of  $n$



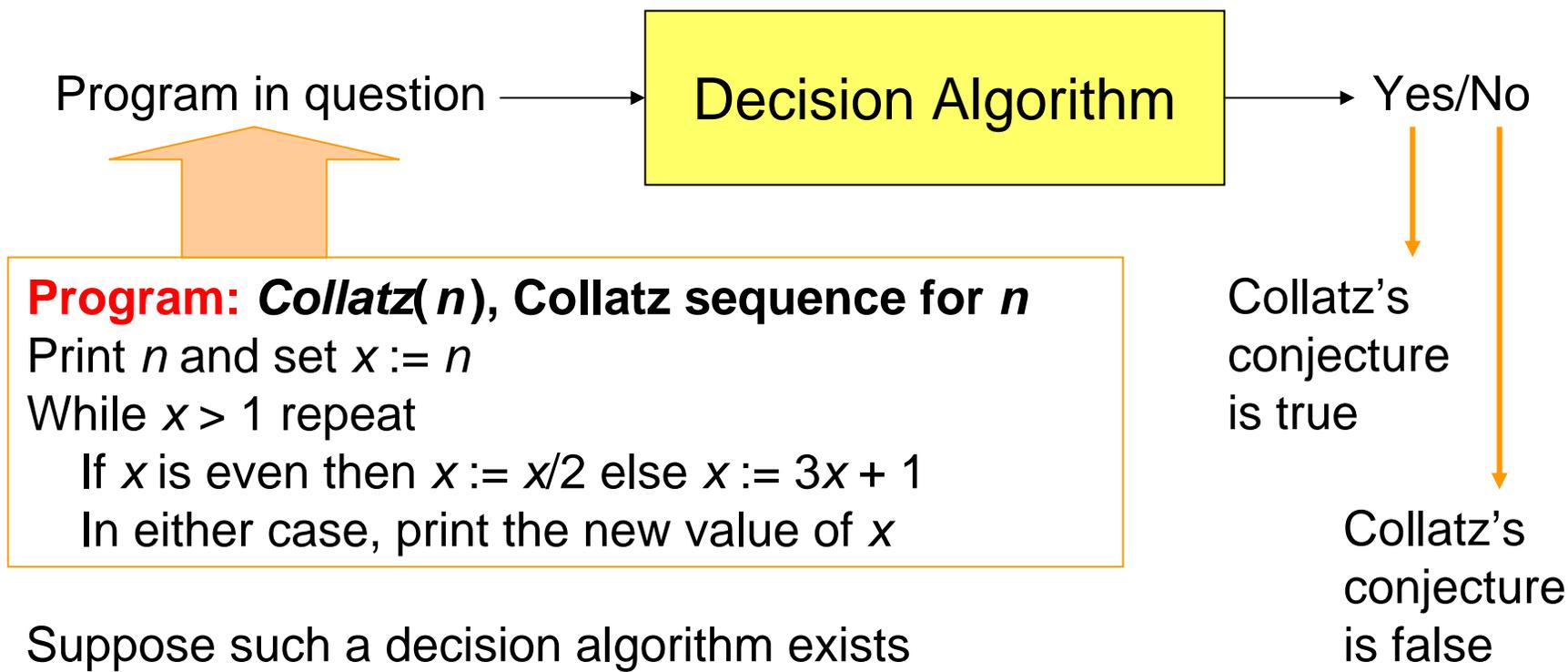
Diagrams from: Weisstein, Eric W., "Collatz Problem," *MathWorld* (<http://mathworld.wolfram.com/CollatzProblem.html>)

In CE, the computer is both an object of study and a tool to help the study

# Are Collatz Sequences Useful?

Not directly, but they help us understand the nature of difficult problems

**Decision problem:** Is there an algorithm to decide whether a given program computing  $f(n)$  will eventually stop for every input  $n$ ?



# Two Other Easy-Looking Hard Problems

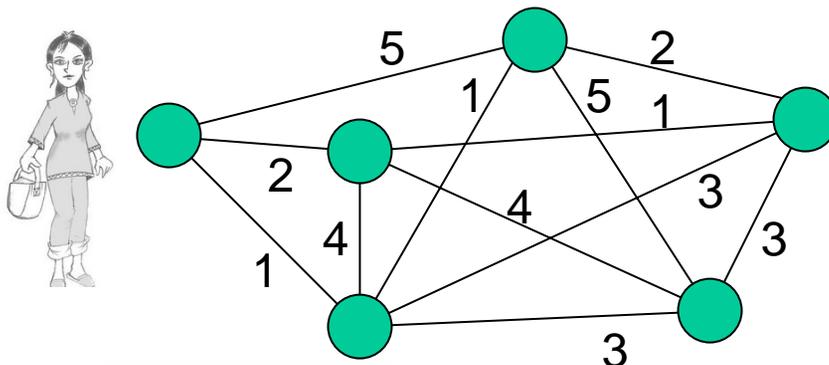
**The subset sum problem:** Given a set of  $n$  numbers, determine whether there is a subset whose sum is a given value  $x$

$$S = \{3, -4, 32, -25, 6, 10, -9, 50\}$$

$$x = 22$$

Can't do fundamentally better than simply trying all  $2^n$  subsets (exponential time, intractable for even moderately large  $n$ )

**The traveling salesperson problem:** Given a set of  $n$  cities with known travel cost  $c_{ij}$  between cities  $i$  and  $j$ , find a path of least cost that would take a salesperson through all cities, returning to the starting city



In the worst case, must examine nearly all the  $(n-1)!$  cycles, which would require exponential time