This class teaches two things:

- Mathematical maturity
- Theory of computation

## Mathematical maturity

- Is the key to success in your scientific career
- In this class we will:
- Practice mathematical notation: sets, quantifiers, etc.
- Get some exposure to proofs

# Theory of computation

- We develop models of computation, and ask: what can and cannot be computed in these models, and how quickly? with how much memory?

- Questions fundamental to all of science.

  Nature computes!

# Theory of computation

- Most famous open question:

Is $P = NP$?

- "Millennium Problem" with prize $1M

- We will learn what this question means in this class

- Overview of material in Theory of Computation
- Automata Theory
- Computability Theory
- Complexity Theory

- Automata Theory

- Finite automata: Computers with no memory

  Motivation: Numbers, names, in Prog. Languages

  Example: *x = -0.0565*

- Context-free grammars: Memory = stack

  Motivation: Syntax, grammar of Prog. Languages

  Example: *if (…) then (…) else (…)*

- Computability Theory

- Turing Machines: “Compute until the sun dies”

- Motivation: What problems can be solved at all?

- Example: Given a program, does it have a bug?

We will prove impossible to determine!

- Complexity Theory

- P, NP: Model your laptop

- Motivation: What problems can be solved fast?

- Example: Given a formula with 1000 variables like

(X OR Y) AND (X OR NOT Z) AND (Z OR Y) …

Is it satisfiable or not?

Does it take 1 thousand years or 1 second to know?

- Recap

- Automata theory: Finite automata, grammars

- Computability Theory: Turing Machines

- Complexity Theory: P, NP

- Theme: Computation has many guises:

  Automata, grammar, Turing Machine, formula ...