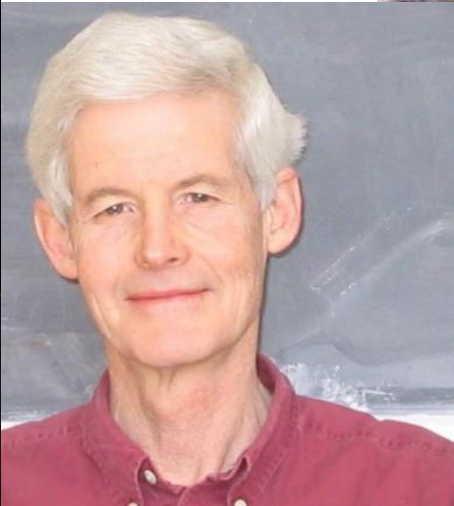
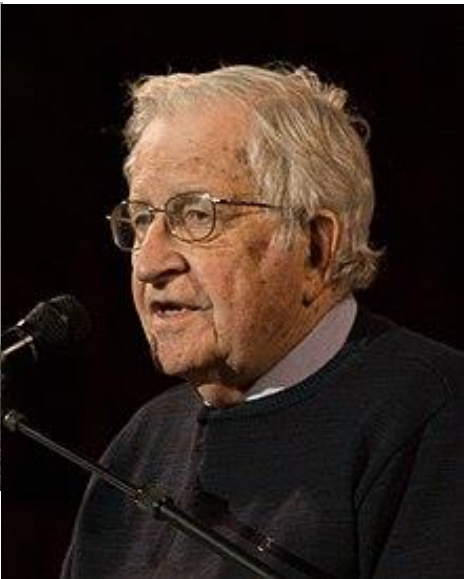
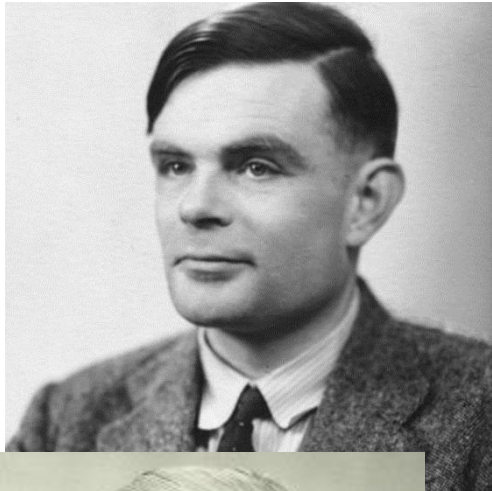


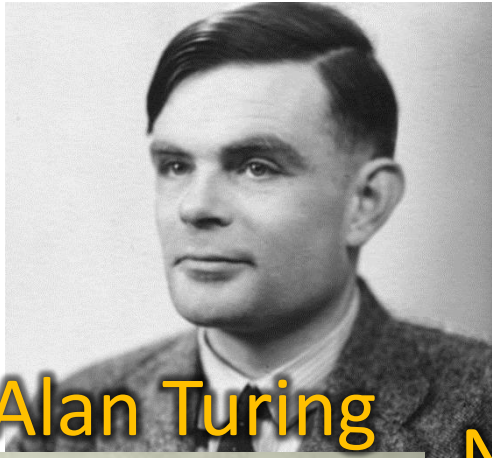
CS3102 Theory of Computation

www.cs.virginia.edu/~njb2b/cstheory/s2020



CS3102 Theory of Computation

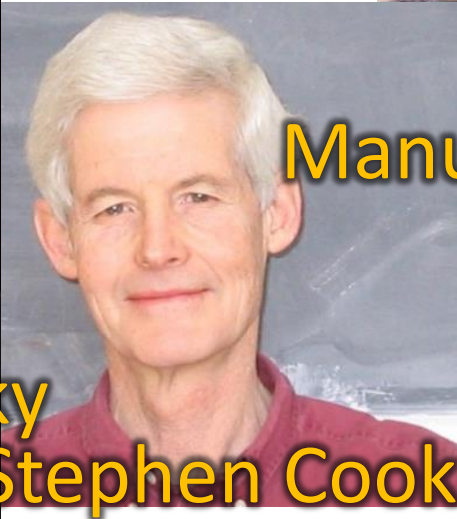
www.cs.virginia.edu/~njb2b/cstheory/s2020



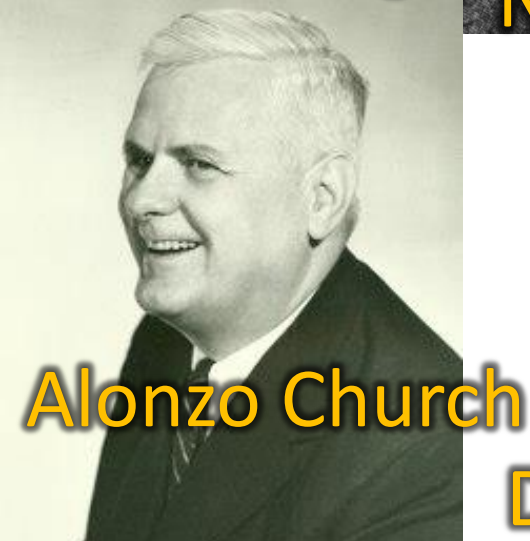
Alan Turing



Noam Chomsky



Stephen Cook



Alonzo Church



Donald Knuth



Boaz Barak



Manuel Blum



Leonid Levin



David Hilbert



Ada Lovelace



Nate Brunelle

Why Study Theory?

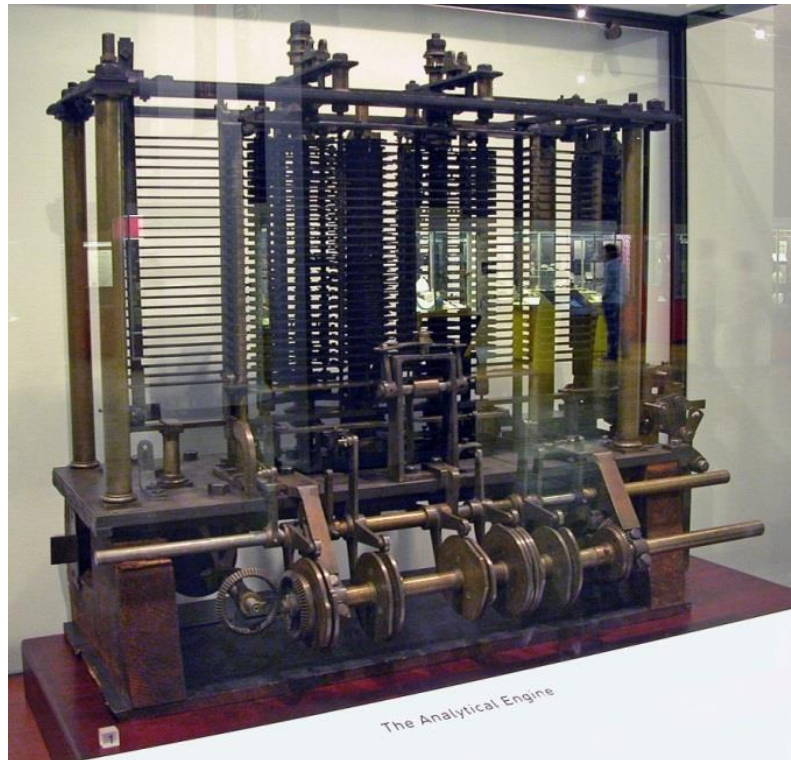
- Foundational knowledge of the field
- A way to clearly communicate ideas
- Develop a general “computational thinking”
- What is or isn’t attainable
- Connect CS with other disciplines
- Gives a framework for considering questions

Consider a Mayan Astronomer





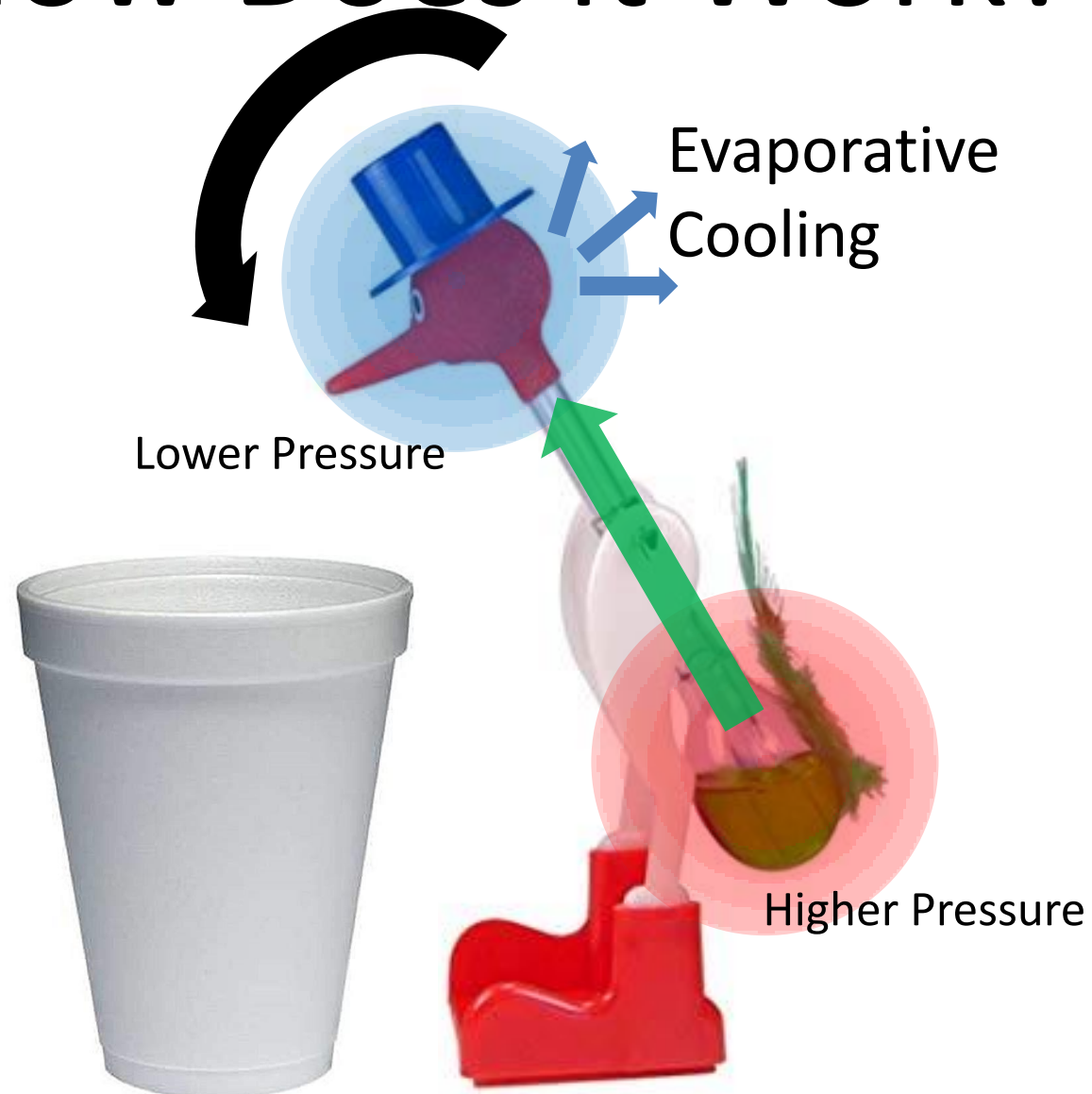
Computers



How Does it Work?



How Does it Work?

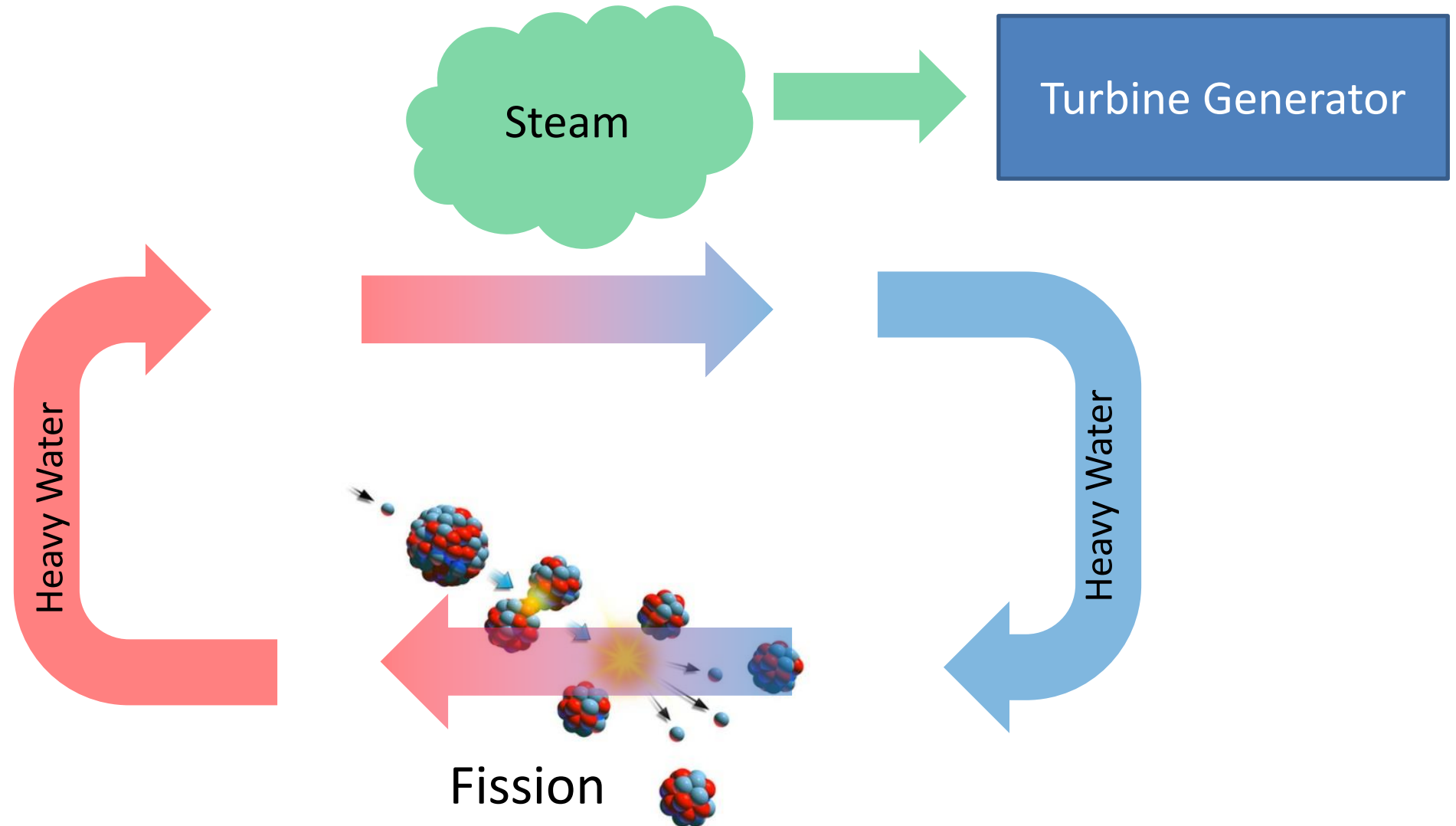


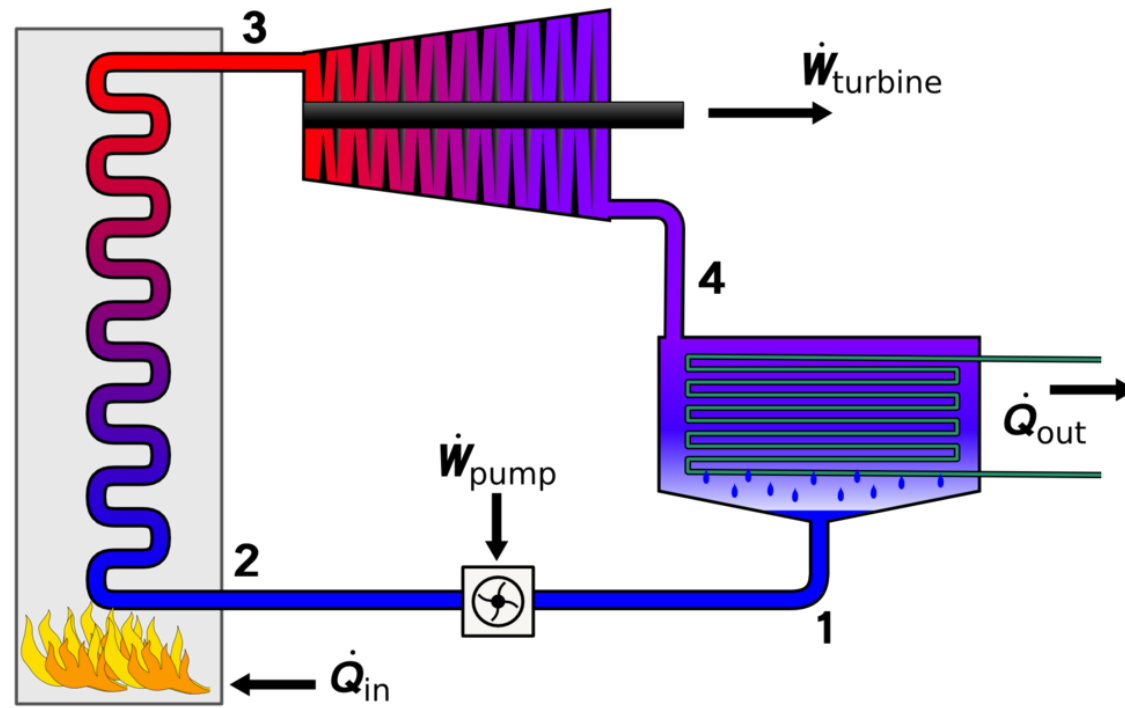
How would I power Charlottesville with a drinking bird?

- Double-headed bird
- More birds!
- Big bird
- Different liquid inside (novac 1230)
- Narrow straw
- Something besides water to evaporate
- Remove the hat
- Add flames/racing stripes, or a spoiler
- Circulating water source
- Define Charlottesville to have 0 population
- lubricant



How does a nuclear power plant work?



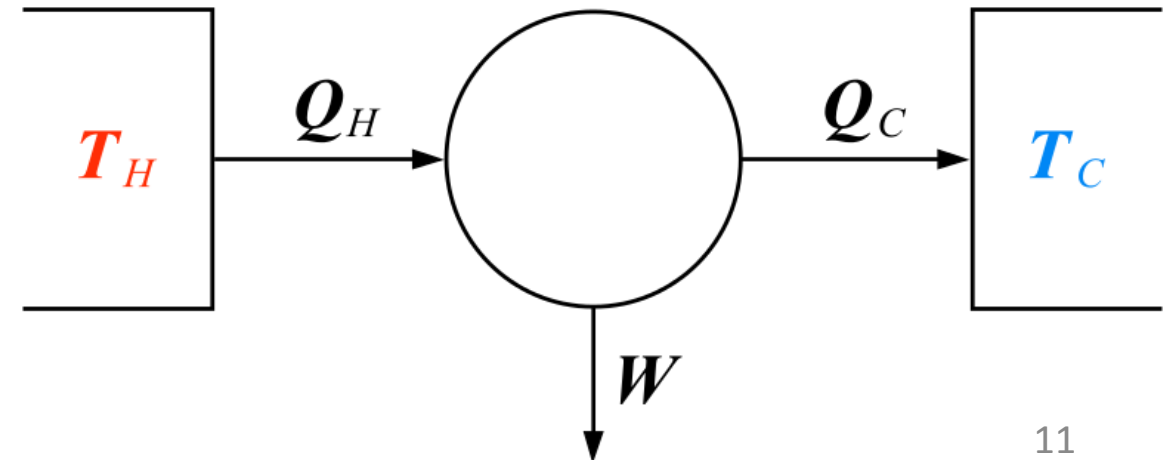


Carnot Engine

- Model of *any* heat engine
- Independent of specifics of construction
- Provides fundamental limits on efficiency

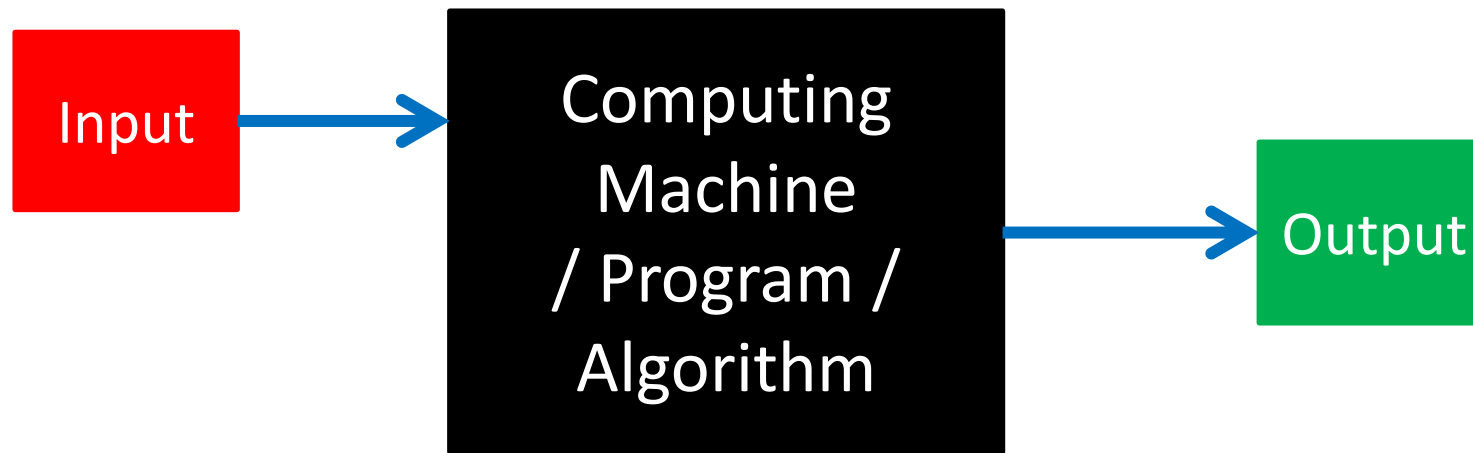


Nicolas Lèonard Sadi Carnot
(1796-1832)



“Carnot Engine” for computers?

- General enough to describe *any* computation
- Independent of specifics of construction
- Enable discussion of limits of computability



What goes in here?

Goals

- Write convincing arguments using formal definitions and mathematical reasoning.
- Reason about the differences between finite and infinite models of computation and what they can and cannot compute.
- Express intuitively and formally what makes some problems too expensive to solve, and what can be done in practice when an unsolvable or intractable problem is encountered.
- Reason formally about the cost of computation, and be able to prove useful bounds on the costs of solving problems, including showing that certain problems are intractable.

Warning

- This may be uncomfortable
 - Material can be subtle
 - Focused on making strong/elegant/compelling arguments
- Lots of opportunities to succeed!

Office Hours

- Nate's
 - Rice 209
 - Mondays, Wednesdays 3:30pm-5:30pm
- TA
 - TBD

Requirements

- Discrete Math (CS 2102)
- Software Development Methods (CS 2110)
- Tenacity
- Inquisitiveness
- Creativity

Text

Boaz Barak, *Introduction to Theoretical
Computer Science*

introtcs.org

Tasks

- Exercises
- Quizzes
- Exams

Survey

- Due Thursday, January 16, 5:30pm
- Includes a short reading
- I expect it will take 20 minutes