Theoretical *vs.* Empirical Analysis

Ideal: Analytically prove properties of a given algorithm (run-time: worst-case / average-case / distribution, error rates).

Reality: Often only possible under substantial simplifications or not at all.

$\rightsquigarrow$ Empirical analysis

The Three Pillars of CS:

- **Theory:** abstract models and their properties
  (“eternal thruths”)
- **Engineering:** principled design of artifacts
  (hardware, systems, algorithms, interfaces)
- **(Empirical) Science:** principled study of phenomenae
  (behaviour of hardware, systems, algorithms; interactions)

The “S” in CS – Why CS is a Science

Definition of ”science”:

(according to the Merriam-Webster Unabridged Dictionary)

“3a: knowledge or a system of knowledge covering general truths or the operation of general laws especially as obtained and tested through scientific method”

(Interestingly, this dictionary lists “information science” as well as “informatics”, but not “computer science”.)

Why “Computer Science” is a Misnomer:

CS is not a science of *computers* (in the standard sense of the meaning), but a science of *computing and information*.

CS is concerned with the study of:

- mathematical structures and concepts that model computation and information (theory, software)
- physical manifestations of these models (hardware)
- interaction between these manifestations and humans (HCI)

The Scientific Method

make observations

formulate hypothesis/hypotheses (model)

While not satisfied (and deadline not exceeded) iterate:

1. design experiment to falsify model
2. conduct experiment
3. analyse experimental results
4. revise model based on results