Motivation

Math analysis is exact, general in nature, and helpful in focusing on the algorithm, but is less readily applicable in some cases.

What if I can't do the math?

- ✎ Math too difficult
- ✎ Math skills not readily available

Accessible, more applicable, but involves opportunities for mis-steps leading to false results.

Empirical analysis: design an experiment

Design is about identifying **requirements**, evaluating **choices**, and making **decisions**.

Effective design requires

- ✎ Focus on important requirements
- ✎ Knowledge of potential choices
- ✎ Experience with relevant decisions
- ✎ Proficiency with tools needed to realize design

Empirical Analysis
🕮 A General Plan

➪ Instrumentation code

Mistakes in 1,3 can undermine the study and may lead to useless results regardless of how well other steps were performed or how much work was involved.

1. Formulate question: experiment goals
2. Choose what to measure: metrics
3. Decide input sampling strategy
4. Prepare program (experiment code)
5. Generate inputs: the test cases
6. Run code + record results
7. Analyze results

Quiz 🕮
What's the principal strength of empirical analysis of algorithms?

Empirical Analysis Poject
Where Do I Begin?*

Check the grading rubric for a guide to good results.

➪ **Read assignment carefully**

There is an official project topic in the course group.

➪ **Ask questions in course group**

Read the first few pages, then reference rest as needed.

➪ **Start with Section 2.6**

Like choosing the right map for a trip, mistakes here can lead to wrong results.

➪ **Think over decisions (1-3)**

* This class was reformatted based on this simple question posed by a student from a previous semester.

A Note about Metrics

A beginner may think that physical runtime has no place in dealing with algorithm performance.

Is physical run time useless?

Algorithm, an abstract idea, may be expressed in many ways from mathematical to free-form natural language.

Logical vs physical algorithm

- Human representations

  Flowcharts, computer-like pseudocode, or machine-readable **computer program** in a programming language like C.

- Machine code

  Internal physical 0s and 1s, as much a reflection of physical hardware in the machine as is of the logical solution.

Operation count or runtime?

Basic operation counts focus on the algorithm. Physical runtime is useful for program performance (whose backbone is algorithm efficiency among other components).

When an efficient algorithm performs poorly on a certain machine, focus turns to physical time to figure out the reasons.

Quiz
Which metric best compares recursive and nonrecursive algorithms performing the same basic operations?