Analysis of Algorithm

Key: Running Time (first considered by Charles Babbage)  
  
Reasons to analyze algorithm:

* Predict performance
* Compare algorithms
* Provide guarantees
* Understand theoretical basis

Primary practical reason: avoid performance bugs  
  
Some algorithm successes

* + FFT algorithm (Fast Fourier Transform): break down waveform of N samples into periodic components  
    Applications: DVD, JPEG, MRI, astrophysics, ...  
    Brute Force (the easy way): N^2 steps  
    FFT algorithm: N log N steps, enables new technologies
  + Barnes-Hut algorithm (N-body simulations): simulates gravitational interactions among N bodies  
    Brute Force: N^2 steps  
    Barnes-Hut algorithm: N log N, enables new technologies

The Challenge: Will my program be able to solve a large practical input?  
  
Scientific method applied to analysis of algorithms:

* Observe some features of the natural world
* Hypothesize a model that is consistent with observations
* Predict events using the hypothesis
* Verify the predictions by making further observations.
* Validate by repeating until the hypothesis and observations agree

Principles:

* Experiments must be reproducible
* Hypotheses must be falsifiable