CSE332: Data Abstractions

Lecture 27: A Few Words on NP

Tyler Robison

Summer 2010

Easily one of the most important questions in Computer Science:

Does P=NP?

- Of course, we need to go into what these terms mean
- P and NP are *classes* of problems
  - P: Class of problems that can be solved in polynomial time
  - NP: Class of problems where an answer can be *verified* in polynomial time
    - We'll get into what that means
- The question is, are these sets equivalent?
  - A question that computer scientists & mathematicians have been grappling with for a long time
  - Most believe that P!=NP, but no one's proven it
  - One such proof recently in the news (P!=NP; probably not valid)

Wow, that's fantastic… who cares?

- P=NP would mean that many 'difficult' problems that could previously only be solved in exponential time could now be solved in polynomial time
  - Some algorithms (such as cryptography) are based around the 'difficulty' of brute-forcing it, but the ease of which an answer can be verified
  - You can break many online encryptions now… with enough computing power
    - Say, an enormous # of computers
    - Or one computer running for several centuries
    - And you don't break the scheme itself; you break it for a single session
  - If P=NP, much of existing cryptography would (in theory) be insecure