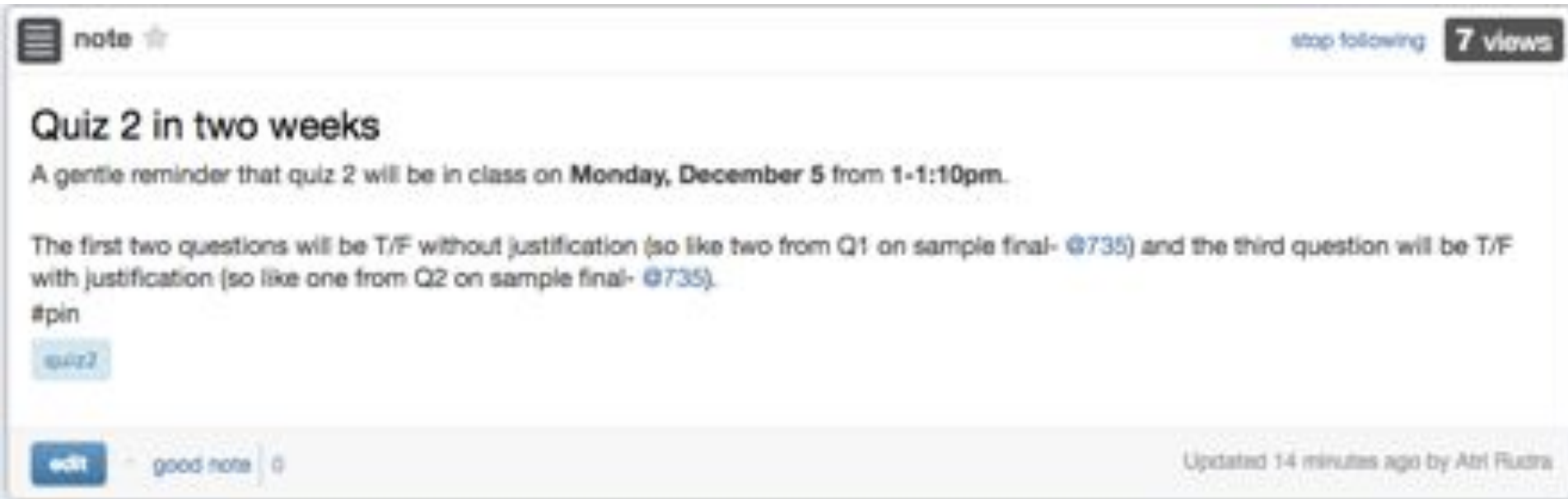


# Lecture 37

CSE 331

Dec 2, 2016

# Quiz 2 on Monday



The image shows a screenshot of a social media post. At the top left, there is a menu icon, the word "note", and a star icon. At the top right, there is a "stop following" button and a "7 views" badge. The main text of the post is titled "Quiz 2 in two weeks" and contains a reminder about the quiz date and time, followed by details about the questions and a hashtag. At the bottom left, there is a "good note" button and a comment icon. At the bottom right, there is an update timestamp and the user's name.

note ☆

stop following 7 views

## Quiz 2 in two weeks

A gentle reminder that quiz 2 will be in class on **Monday, December 5** from 1-1:10pm.

The first two questions will be T/F without justification (so like two from Q1 on sample final- @735) and the third question will be T/F with justification (so like one from Q2 on sample final- @735).

#pin

quiz2

good note 0

Updated 14 minutes ago by Atri Ruzra

You can use two letter sized cheatsheets

# Last HW up!

## Homework 10

Due by **12:30pm, Friday, December 9, 2016.**

Make sure you follow all the [homework policies](#).

All submissions should be done via [Autolab](#).

### Question 1 (Programming Assignment) [40 points]

#### `</>` Note

This assignment can be solved in either Java, Python or C++ (you should pick the language you are most comfortable with). Please make sure to look at the supporting documentation and files for the language of your choosing.

#### The Problem

In this problem, you are given a directed graph (in adjacency list representation)  $G = (V, E)$  where each edge  $e \in E$  has cost  $c_e$  (which can be negative but  $G$  does

# HW 9 solutions

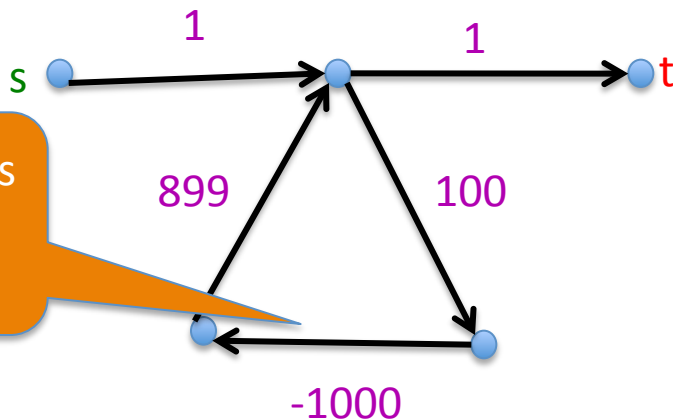
At the END of the lecture

# Shortest Path Problem

Input: (Directed) Graph  $G=(V,E)$  and for every edge  $e$  has a cost  $c_e$  (can be  $<0$ )

$t$  in  $V$

Output: Shortest path from every  $s$  to  $t$



Shortest path has cost negative infinity

Assume that  $G$  has no negative cycle

# When to use Dynamic Programming

There are polynomially many sub-problems



Richard Bellman

Optimal solution can be computed from solutions to sub-problems

There is an ordering among sub-problem that allows for iterative solution

# Sub-problems

$OPT(u,i)$  = cost of shortest path from  $u$  to  $t$  with at most  $i$  edges

# When to use Dynamic Programming

There are polynomially many sub-problems



Richard Bellman

Optimal solution can be computed from solutions to sub-problems

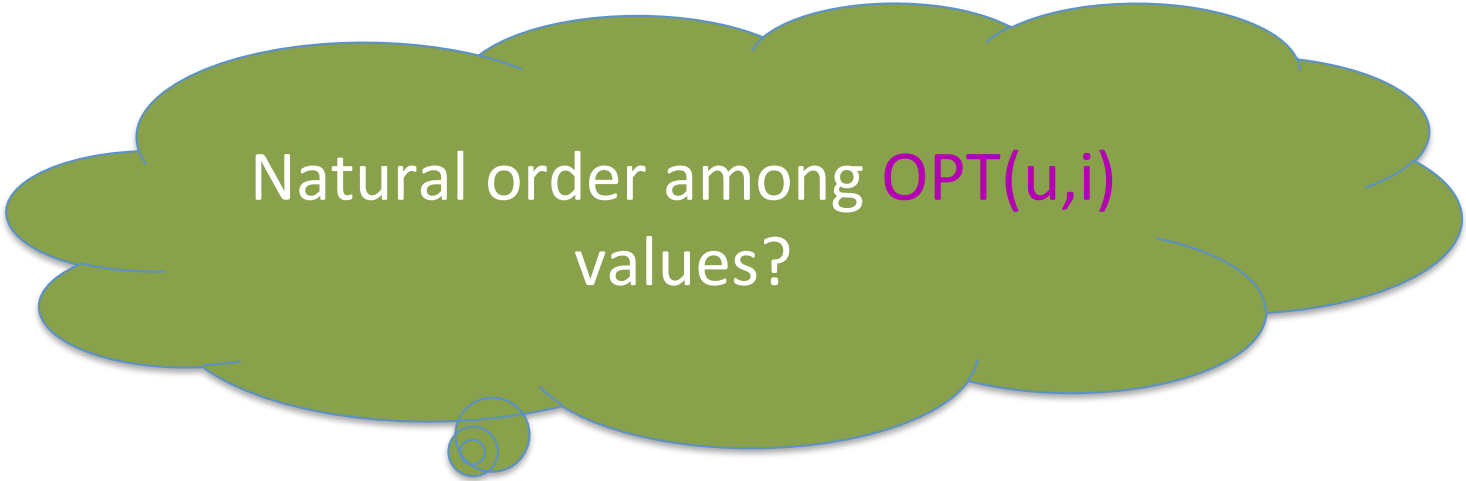
There is an ordering among sub-problem that allows for iterative solution



# Today's agenda

Finish Bellman-Ford algorithm

Analyze the run time



Natural order among  $\text{OPT}(u,i)$   
values?

# The recurrence

$OPT(u,i)$  = shortest path from  $u$  to  $t$  with at most  $i$  edges

$$OPT(u,i) = \min \left\{ OPT(u,i-1), \min_{(u,w) \in E} \left\{ c_{u,w} + OPT(w, i-1) \right\} \right\}$$

# Some consequences

$OPT(u,i)$  = shortest path from  $u$  to  $t$  with at most  $i$  edges

$$OPT(u,i) = \min \left\{ OPT(u, i-1), \min_{(u,w) \in E} \{ c_{u,w} + OPT(w,i-1) \} \right\}$$

$OPT(u,n-1)$  is shortest path cost between  $u$  and  $t$

Group talk time:  
How to compute the shortest  
path between  $s$  and  $t$  given all  
 $OPT(u,i)$  values

# Longest path problem

Given  $G$ , does there exist a simple path of length  $n-1$  ?

# Longest vs Shortest Paths

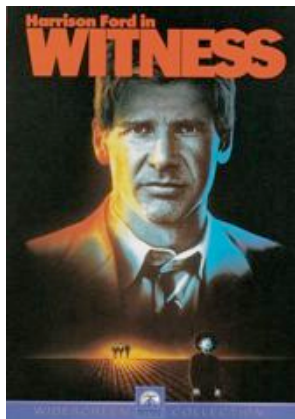


# Two sides of the “same” coin

Shortest Path problem

Can be solved by a polynomial time algorithm

Is there a longest path of length  $n-1$ ?



Given a path can verify in polynomial time if the answer is yes

# Poly time algo for longest path?



## Clay Mathematics Institute

*Dedicated to increasing and disseminating mathematical knowledge*

[HOME](#) | [ABOUT CHI](#) | [PROGRAMS](#) | [NEWS & EVENTS](#) | [AWARDS](#) | [SCHOLARS](#) | [PUBLICATIONS](#)

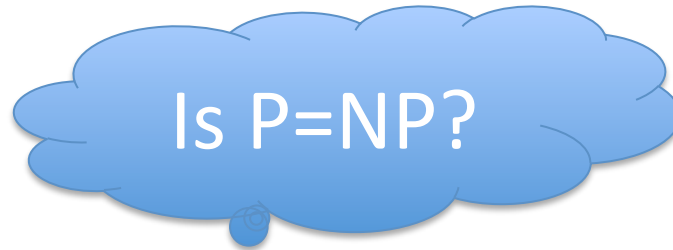
### First Clay Mathematics Institute Millennium Prize Announced

Prize for Resolution of the Poincaré Conjecture Awarded to Dr. Grigoriy Perelman

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- **P vs NP**
- Poincaré Conjecture
- Riemann Hypothesis

# P vs NP question

**P**: problems that can be solved by poly time algorithms



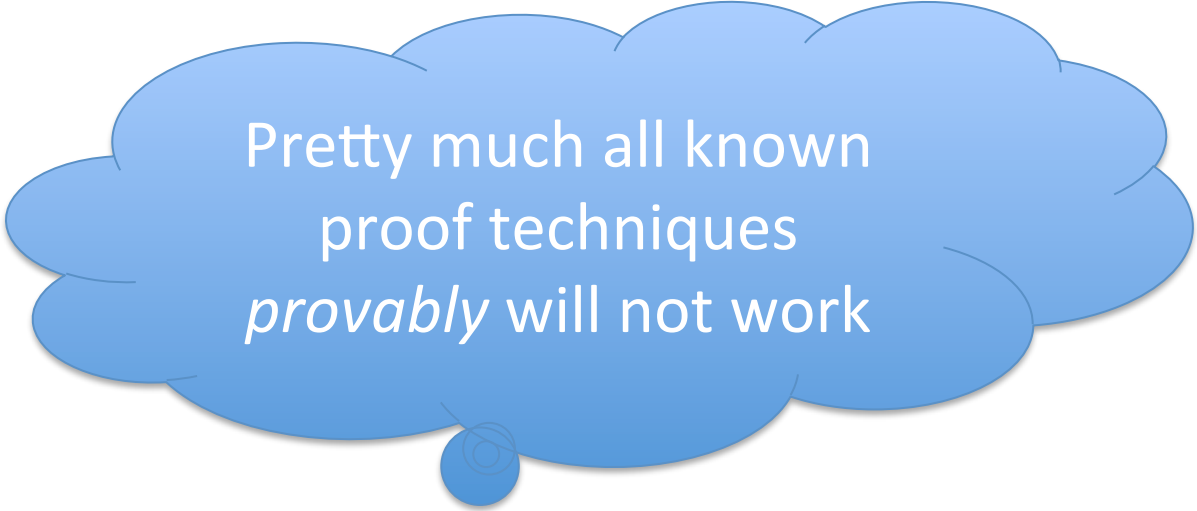
**NP**: problems that have polynomial time verifiable witness to optimal solution

Alternate NP definition: Guess witness and verify!



# Proving $P \neq NP$

Pick any one problem in NP and show it cannot be solved in poly time



Pretty much all known  
proof techniques  
*provably* will not work

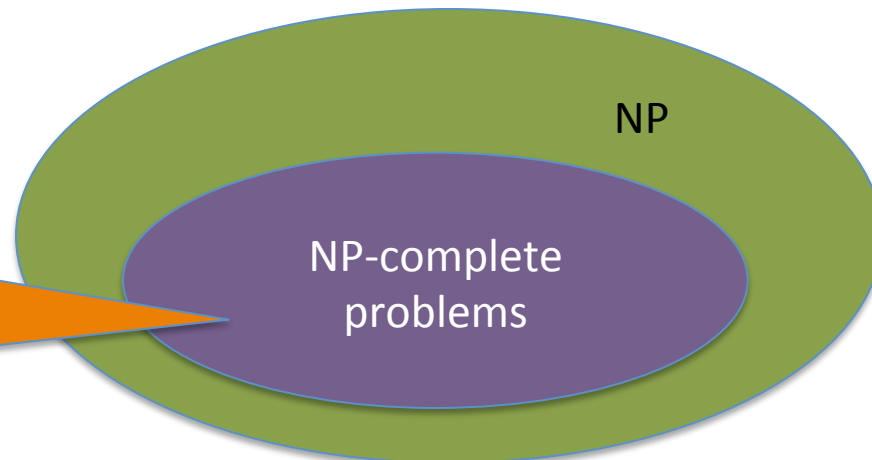
# Proving $P = NP$

Will make cryptography collapse

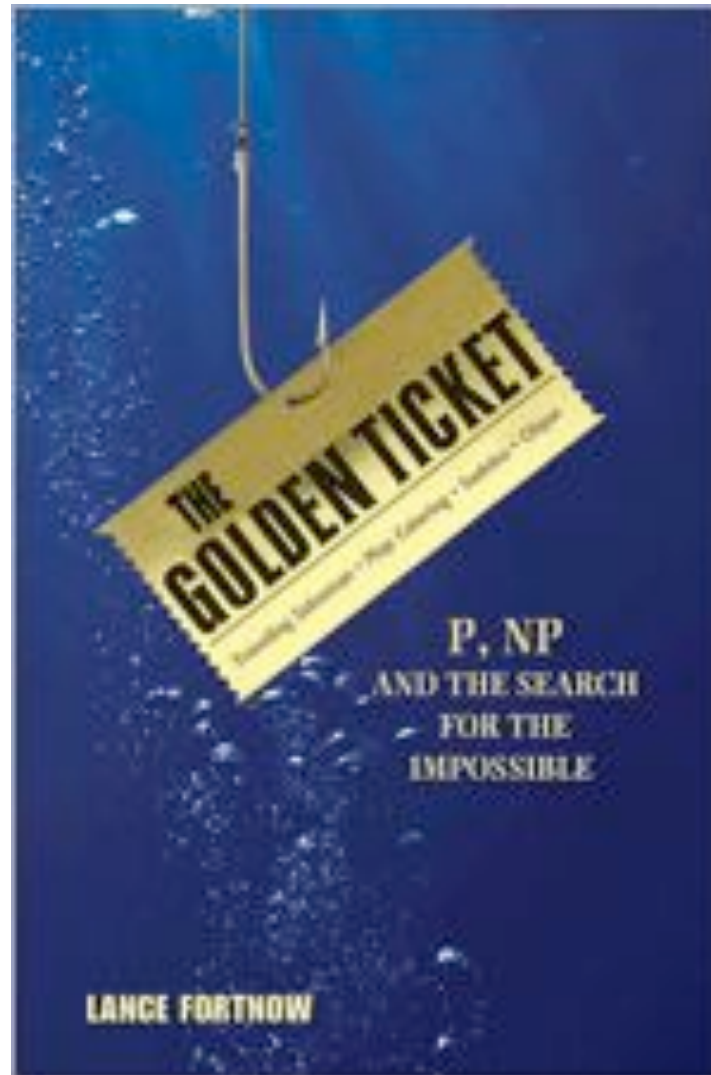
Compute the encryption key!

Prove that all problems in NP can be solved by polynomial time algorithms

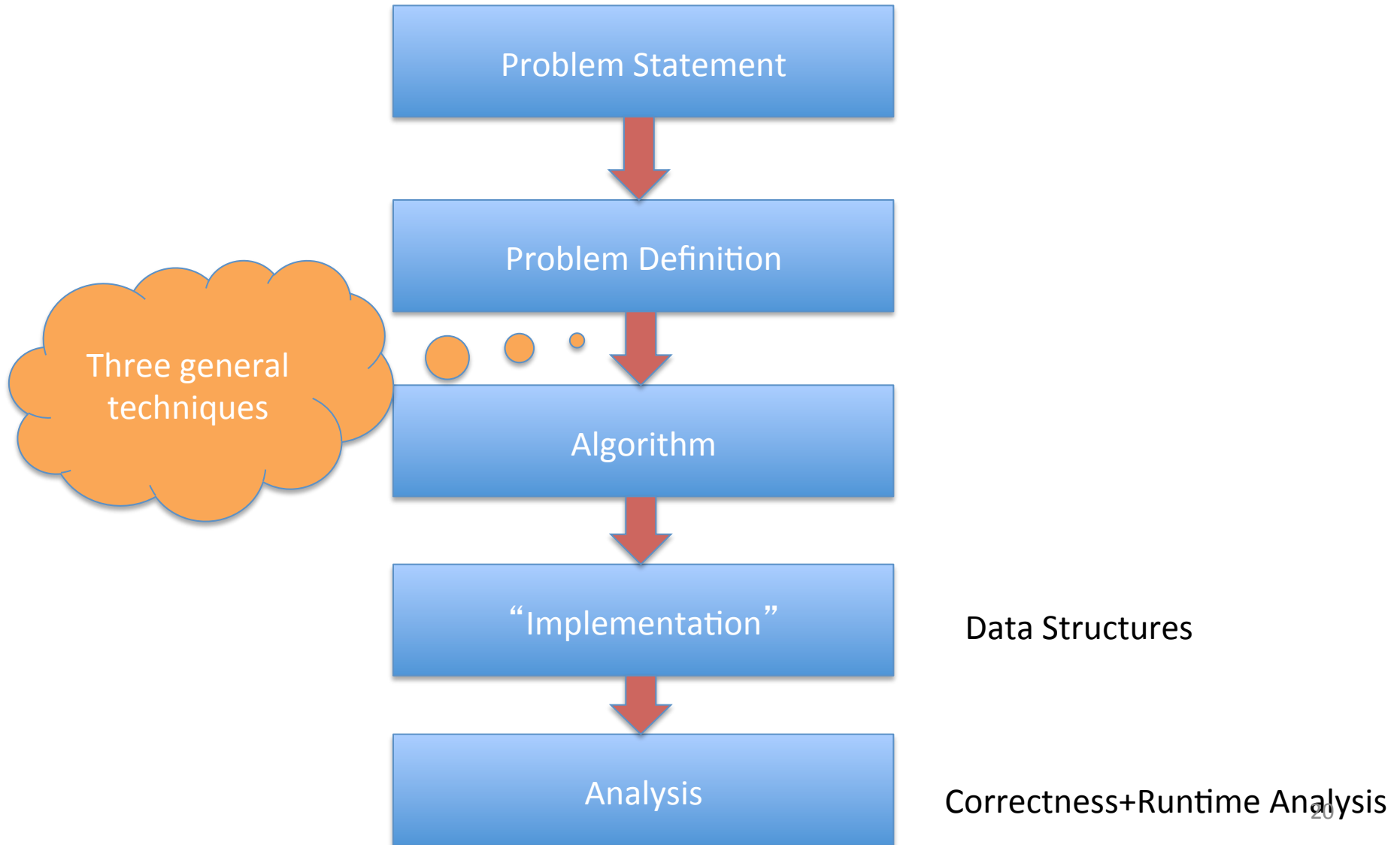
Solving any ONE problem in here in poly time will prove  $P=NP$ !



# A book on P vs. NP



# High level view of CSE 331



# If you are curious for more

CSE 429 or 431: Algorithms

CSE 396: Theory of Computation

