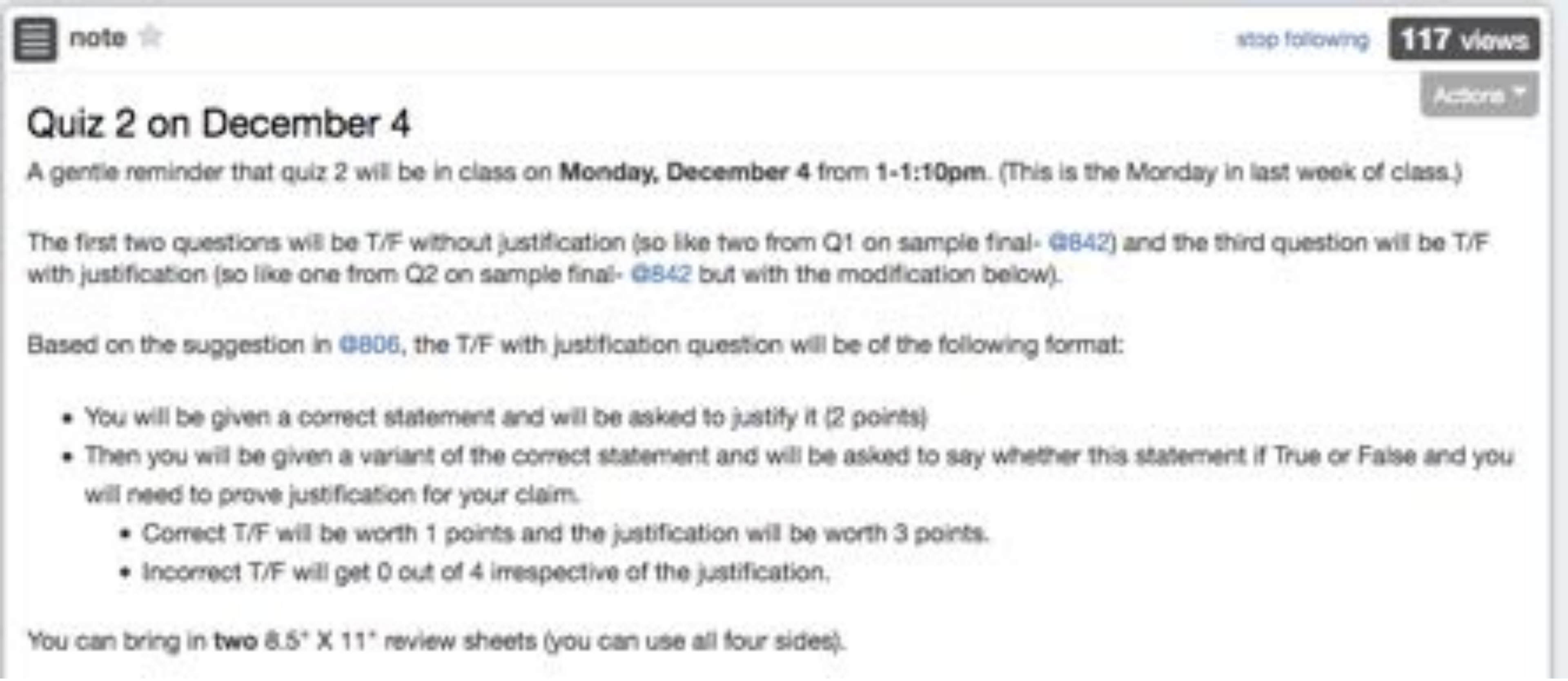
Lecture 37

CSE 331

Dec 1, 2017

Quiz 2 on Monday



Quiz 2 on December 4

A gentle reminder that quiz 2 will be in class on **Monday, December 4** from **1-1:10pm**. (This is the Monday in last week of class.)

The first two questions will be T/F without justification (so like two from Q1 on sample final- @842) and the third question will be T/F with justification (so like one from Q2 on sample final- @842 but with the modification below).

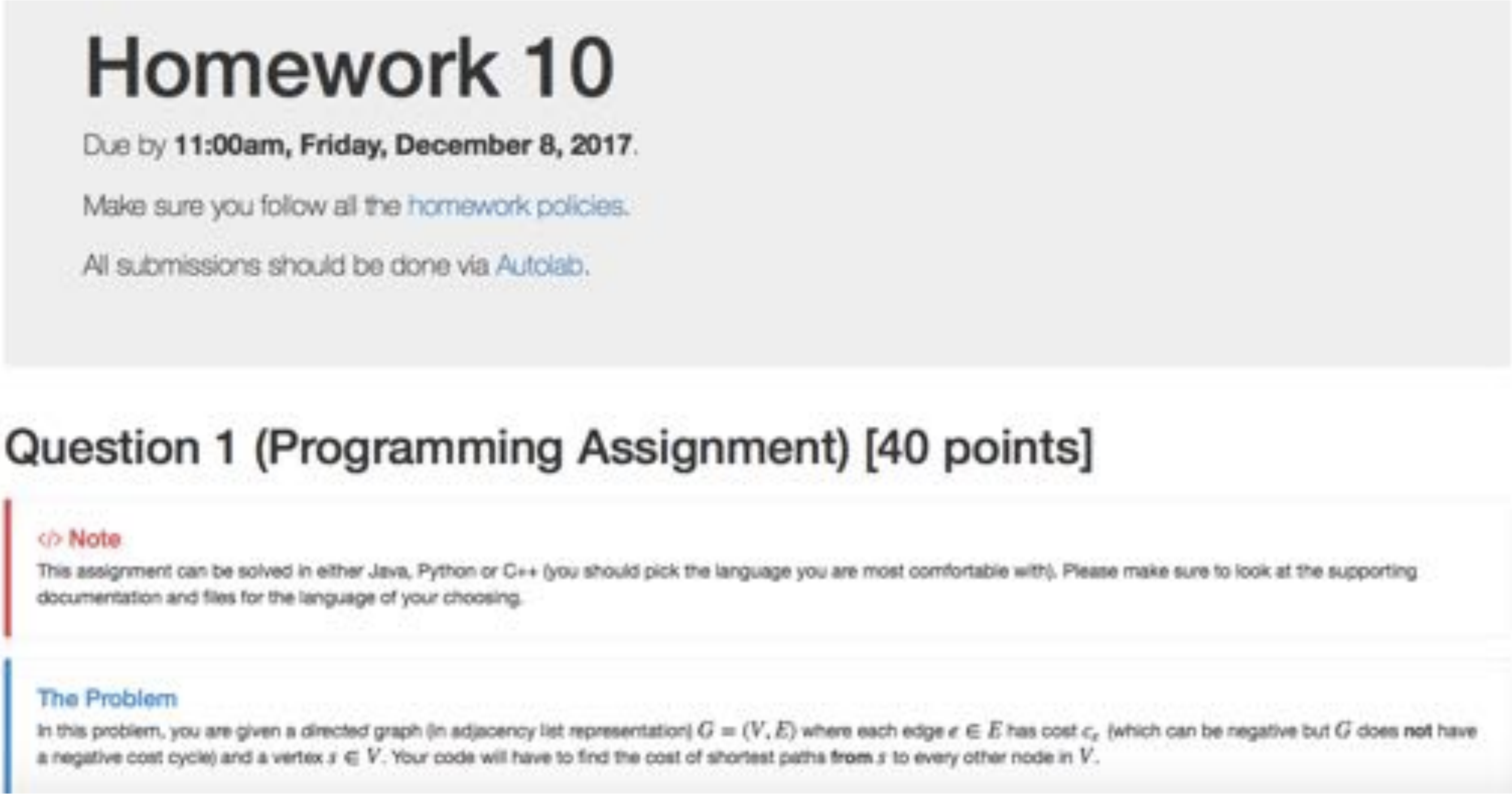
Based on the suggestion in @806, the T/F with justification question will be of the following format:

- You will be given a correct statement and will be asked to justify it (2 points)
- Then you will be given a variant of the correct statement and will be asked to say whether this statement if True or False and you will need to prove justification for your claim.
    - Correct T/F will be worth 1 points and the justification will be worth 3 points.
    - Incorrect T/F will get 0 out of 4 irrespective of the justification.

You can bring in **two** 8.5" X 11" review sheets (you can use all four sides).

You can use two letter sized cheatsheets

Last HW up!



Homework 10

Due by **11:00am, Friday, December 8, 2017**.

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 **not** have a negative cost cycle) and a vertex $s \in V$. Your code will have to find the cost of shortest paths **from** s to every other node in V.

HW 9 solutions

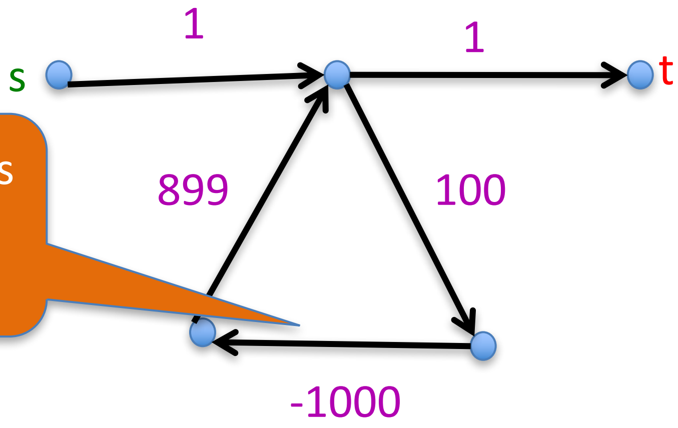
At the END of the lecture

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

Richard Bellman

There are polynomially many sub-problems

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

Today’s agenda

Finish Bellman-Ford algorithm

Analyze the run time

The recurrence

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

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

Some consequences

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

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

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 ?

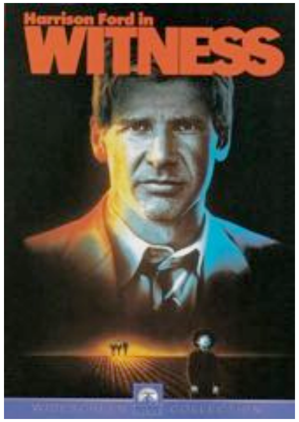
Longest vs Shortest Paths

Two sides of the "same" coin

Shortest Path problem

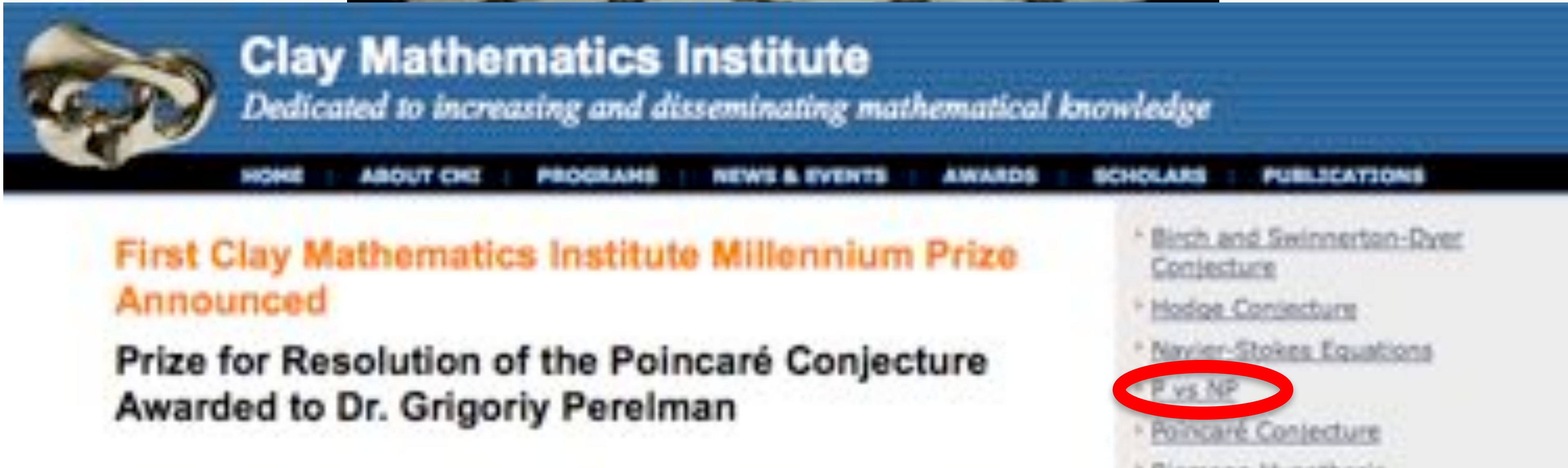
Can be solved by a polynomial time algorithm

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 CMI | 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

Is P=NP?

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

Alternate NP definition: Guess witness and verify!

Proving P ≠ NP

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

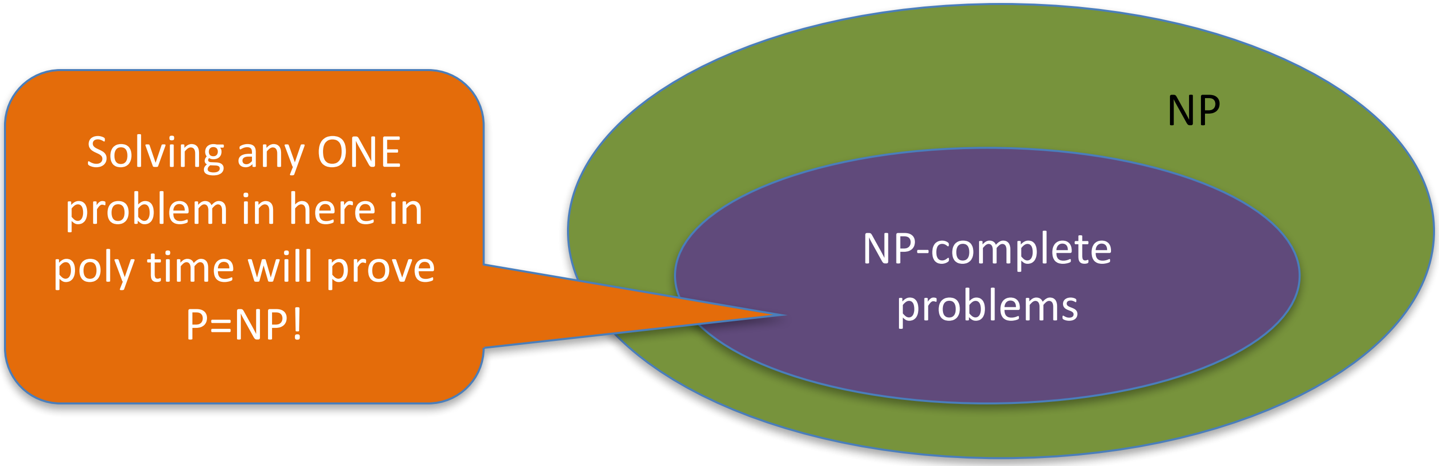
Pretty much all known proof techniques *provably* will not work

Proving P = NP

Will make cryptography collapse

Compute the encryption key!

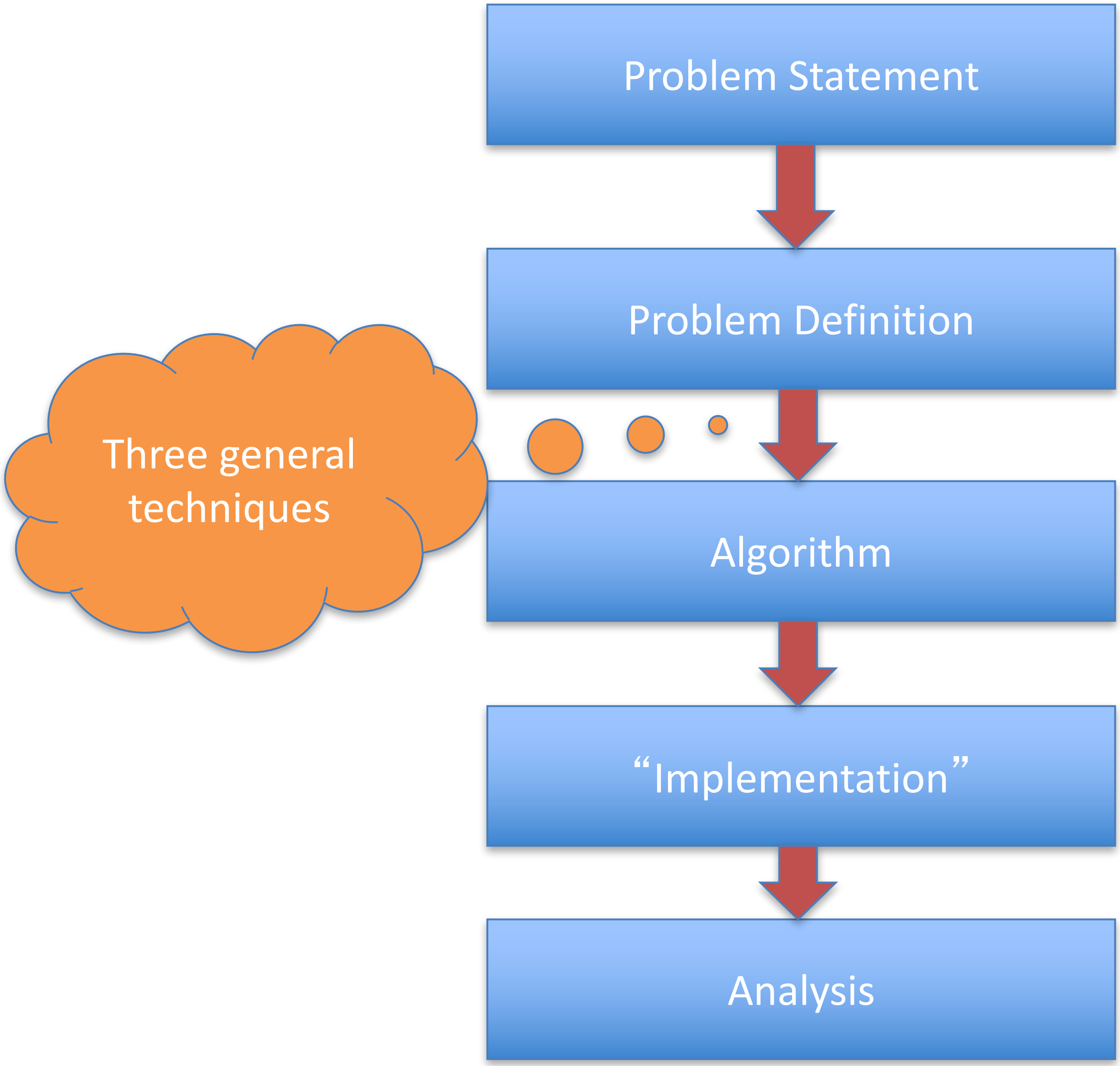
Prove that all problems in NP can be solved by polynomial time algorithms



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



Problem Statement

Problem Definition

Three general techniques

Algorithm

“Implementation”

Analysis

Data Structures

Correctness+Runtime Analysis

If you are curious for more

CSE 429 or 431: Algorithms

CSE 396: Theory of Computation