CSE 332: NP Completeness, Part II

Richard Anderson

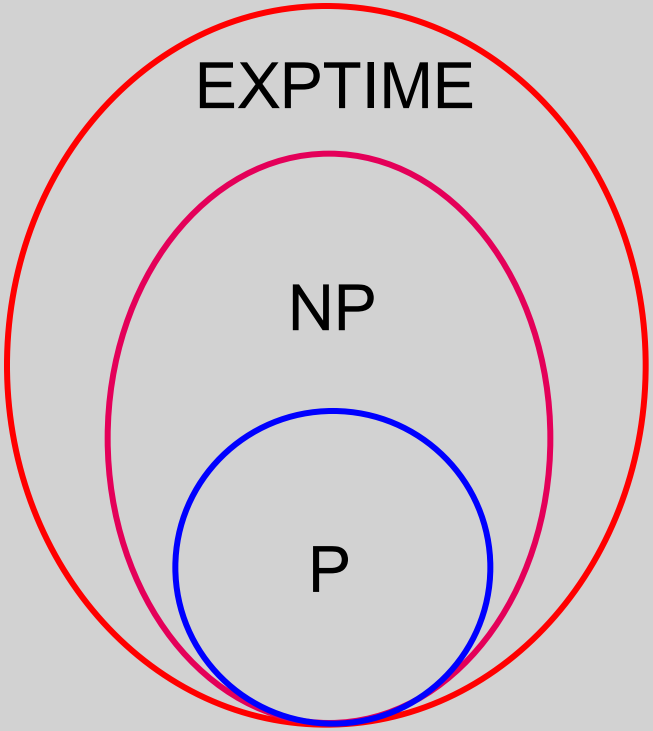
Spring 2016

Announcements

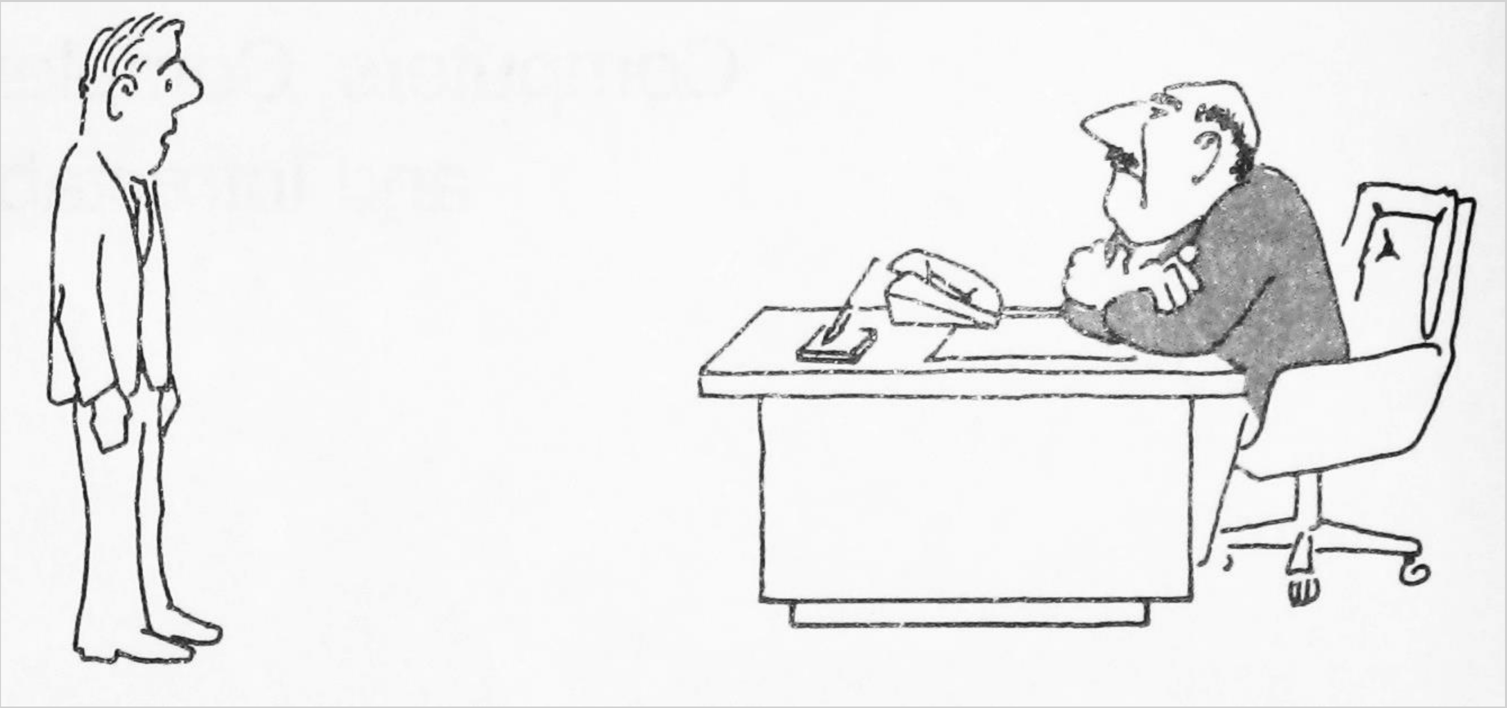
- Review session for the final
  - Friday, in class
- Final
  - Monday, June 6, 2:30-4:20 pm, EEB 037
- Final exam hints
  - About 75% of the exam is post midterm, 25% pre midterm
  - Three questions related to Dijkstra's Algorithm

NP Completeness

- “Easy problems” – solvable in Polynomial Time
- Hard problems – take exponential time
- Interesting class of problems: Non-deterministic polynomial time

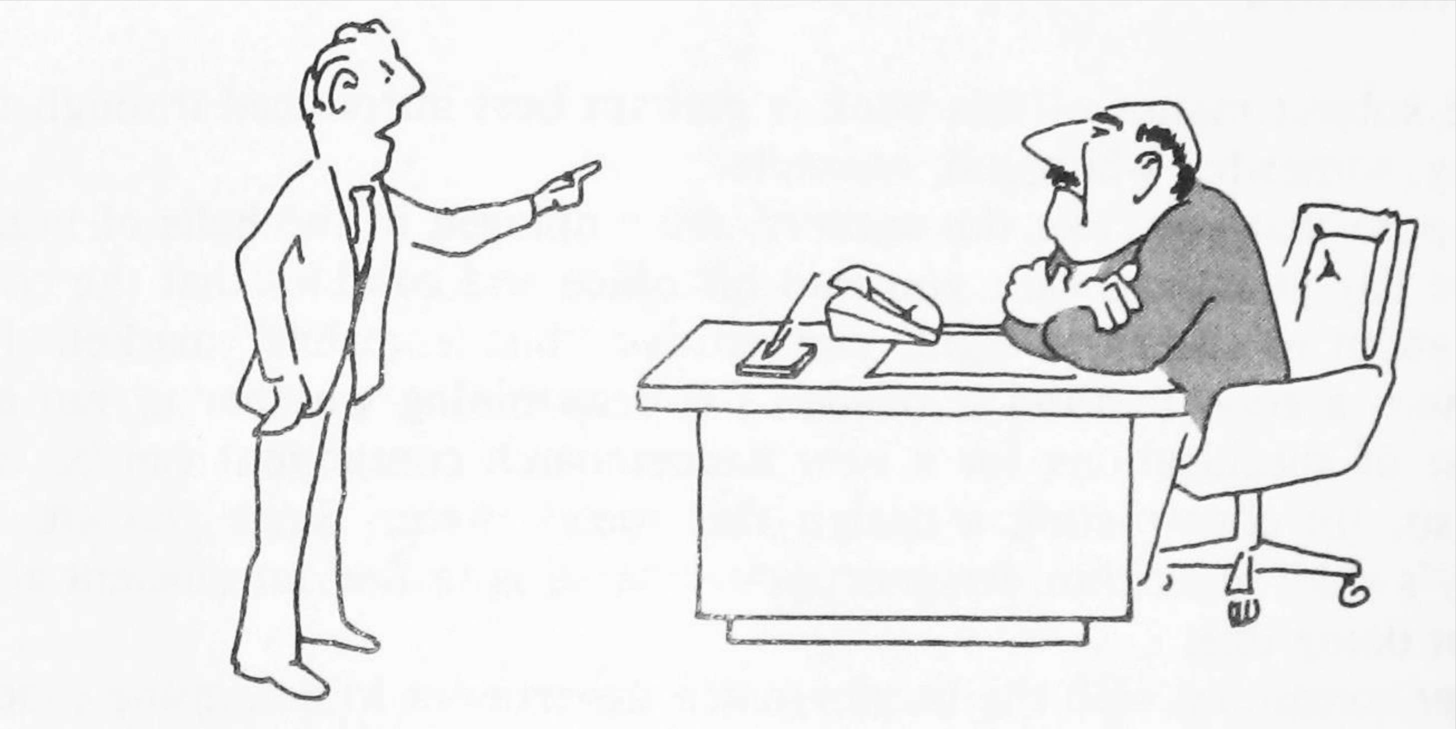


What you'd rather not say…



"I can't find an efficient algorithm, I guess I'm just too dumb."

What you'd like to say…



"I can't find an efficient algorithm, because no such algorithm is possible!"

But can you actually say this…?

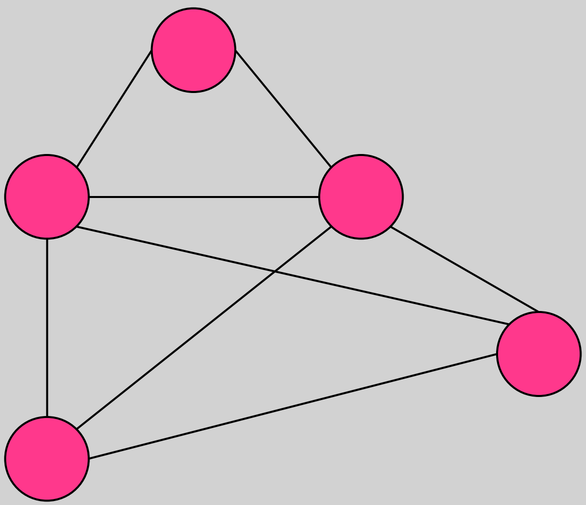
What you *can* say…

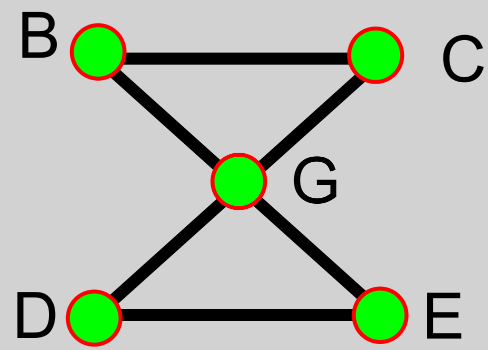
"I can't find an efficient algorithm, but neither can all these famous people."

Problem Reduction

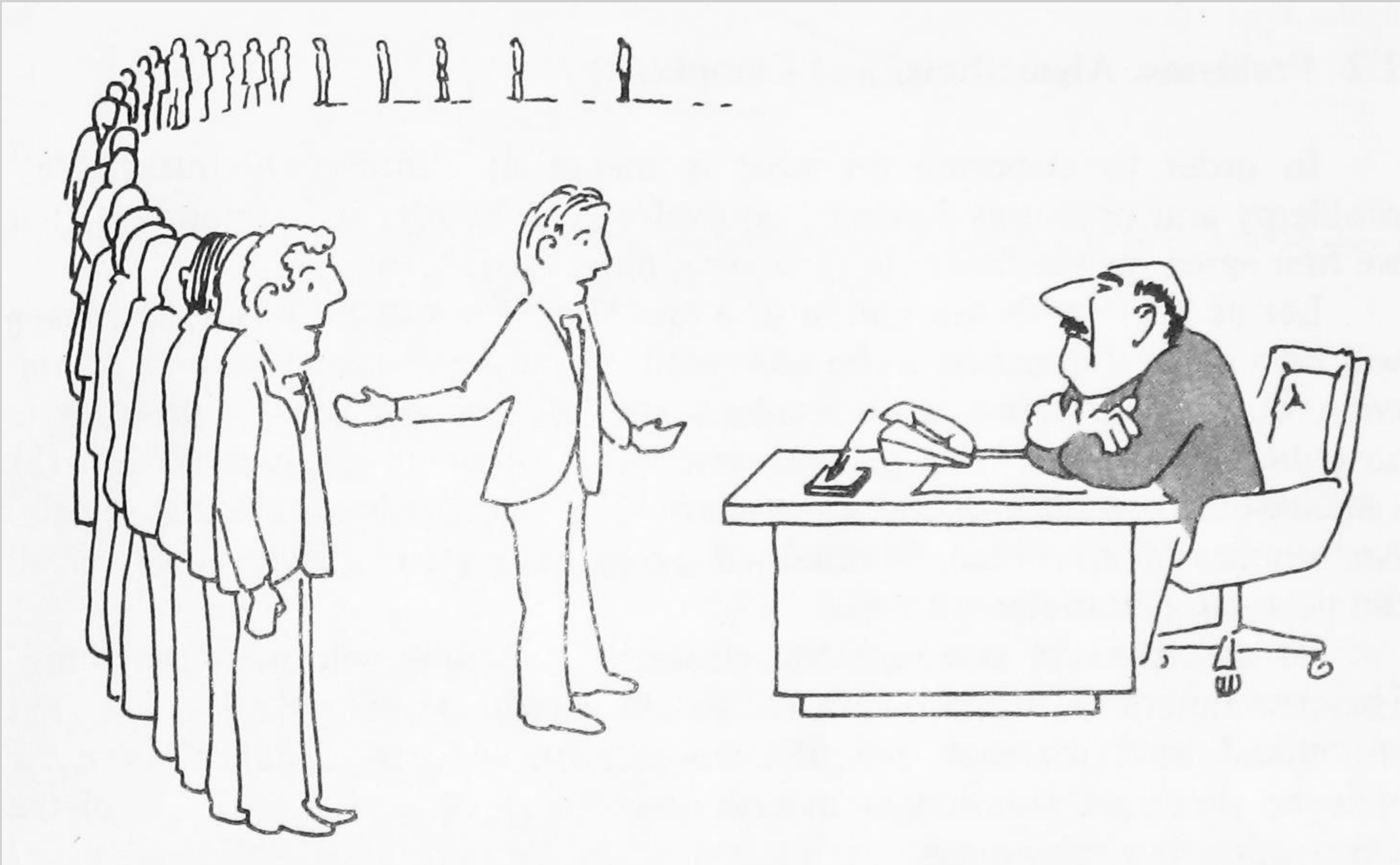
- Hamiltonian Circuit
  - Given a graph, is the a simple cycle that includes all of the vertices
- Travelling Salesman Problem
  - Given a complete graph with edge costs and a constant C, is there a cycle that visits all vertices with total cost at most C
- A reduction of HC to TSP uses an instance of TSP to solve an instance of HP

Examples





What you can say…



"I can't find an efficient algorithm, since all these famous people couldn't solve HC efficiently and I can prove that TSP is at least as hard."

What is NP?

- Problems solvable in non-deterministic polynomial time . . .

- Problems where “yes” instances have polynomial time checkable certificates

Certificate examples

- Independent set of size K
  - The Independent Set
- Satifisfiable formula
  - Truth assignment to the variables
- Hamiltonian Circuit Problem
  - A cycle including all of the vertices
- K-coloring a graph
  - Assignment of colors to the vertices

Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula

Certificate: An assignment of truth values to the n boolean variables

Certifier: Check that each clause has at least one true literal,

instance s

$$\left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(x_1 \vee x_2 \vee x_4\right) \wedge \left(\overline{x_1} \vee \overline{x_3} \vee \overline{x_4}\right)$$

certificate t

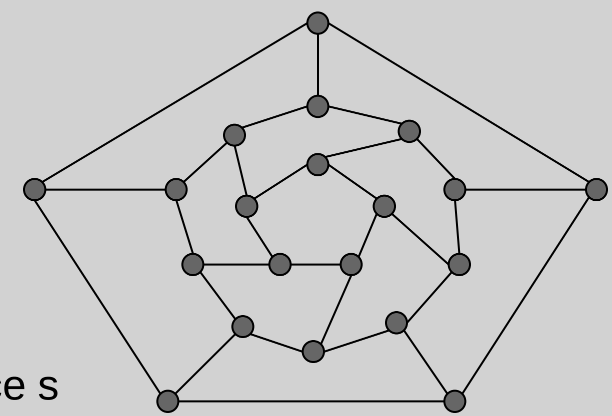
$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$$

Certifiers and Certificates: Hamiltonian Cycle

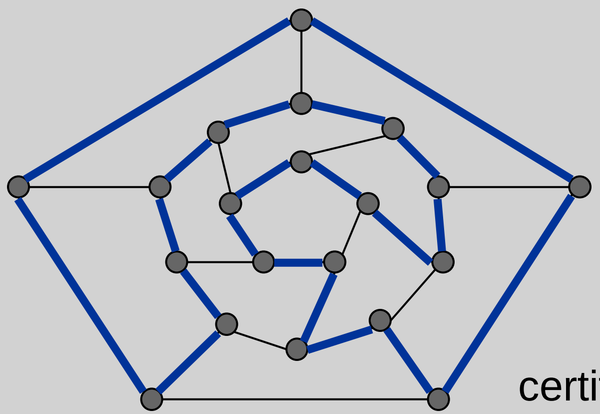
HAM-CYCLE. Given an undirected graph G = (V, E), does there exist a simple cycle C that visits every node?

Certificate. A permutation of the n nodes.

Certifier. Check that the permutation contains each node in V exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.



instance s



certificate t

Polynomial time reductions

- Y is Polynomial Time Reducible to X
  - Solve problem Y with a polynomial number of computation steps and a polynomial number of calls to a black box that solves X
  - Notations: $Y <_P X$

Lemmas

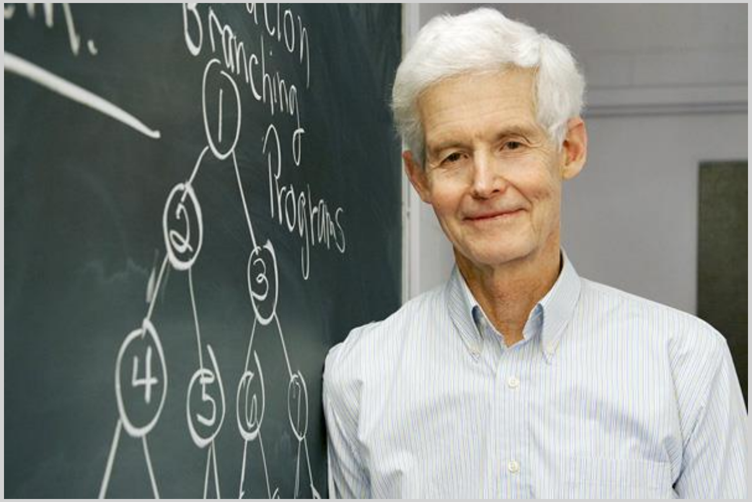
- Suppose $Y <_P X$. If X can be solved in polynomial time, then Y can be solved in polynomial time.
- Suppose $Y <_P X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.

NP-Completeness

- A problem X is NP-complete if
  - X is in NP
  - For every Y in NP, $Y <_P X$
- X is a “hardest” problem in NP
- If X is NP-Complete, Z is in NP and $X <_P Z$
  - Then Z is NP-Complete

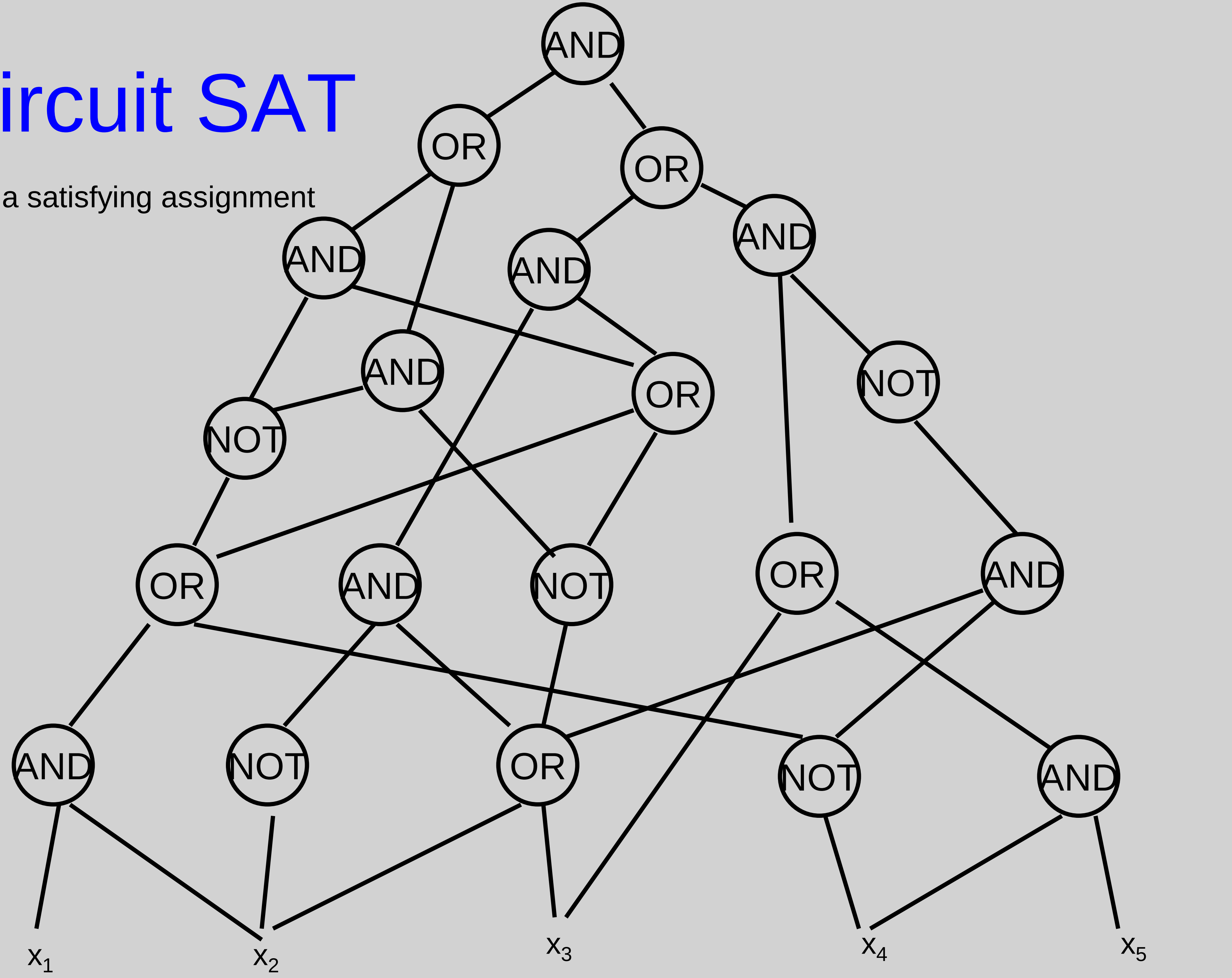
Cook’s Theorem

- The Circuit Satisfiability Problem is NP-Complete



Circuit SAT

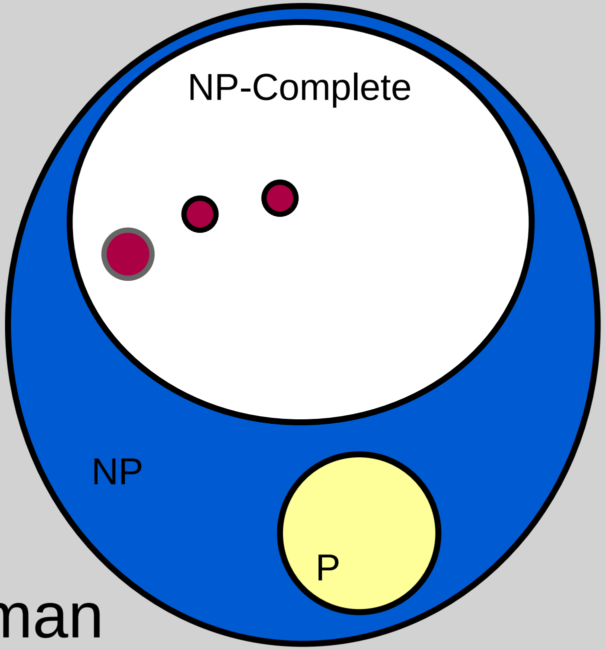
Find a satisfying assignment



Proof of Cook's Theorem

- Reduce an arbitrary problem Y in NP to X
- Let A be a non-deterministic polynomial time algorithm for Y
- Convert A to a circuit, so that Y is a Yes instance iff and only if the circuit is satisfiable

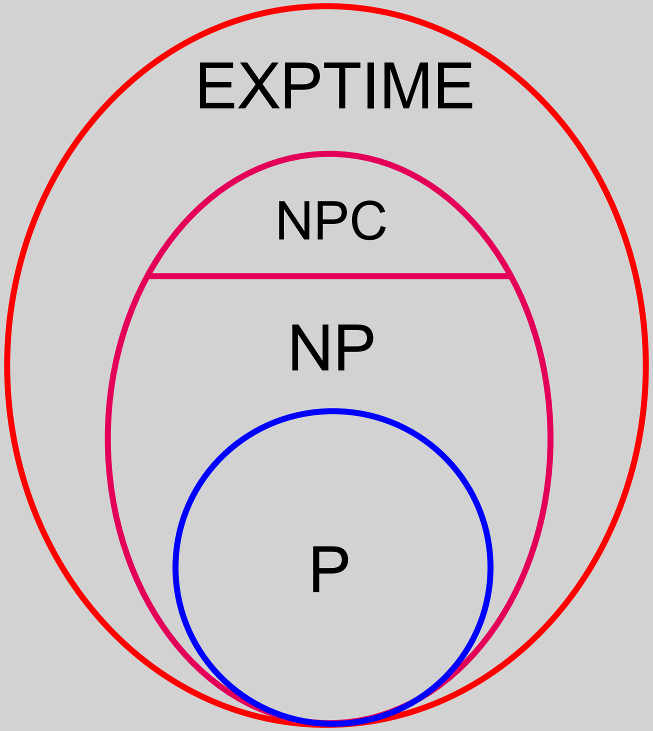
Populating the NP-Completeness Universe



- Circuit Sat $<_P$ 3-SAT
- 3-SAT $<_P$ Independent Set
- 3-SAT $<_P$ Vertex Cover
- Independent Set $<_P$ Clique
- 3-SAT $<_P$ Hamiltonian Circuit
- Hamiltonian Circuit $<_P$ Traveling Salesman
- 3-SAT $<_P$ Integer Linear Programming
- 3-SAT $<_P$ Graph Coloring
- 3-SAT $<_P$ Subset Sum
- Subset Sum $<_P$ Scheduling with Release times and deadlines

P, NP, NPC, and Exponential Time Problems

- All **currently known** algorithms for NP-complete problems run in **exponential** worst case time
- Diagram depicts relationship between P, NP, and EXPTIME (class of problems that **provably require** exponential time to solve)



It is believed that
P ≠ NP ≠ EXPTIME

Great Quick Reference

Is this lecture complete? Hardly, but here's a good reference:

Computers and Intractability:
A Guide to the Theory of
NP-Completeness
by Michael S. Garey and
David S. Johnson