Portland State University

W'21 CS 584/684

Algorithm Design & Analysis

Fang Song

Lecture 17

- Reductions
- P vs. NP

Recall: polynomial-time reduction

- **Def.** Problem X polynomial reduces to Problem Y if arbitrary instance of X can be solved using:
  - Polynomial number of standard computation steps
  - & polynomial number of calls to oracle that solves A

Notation. $X \leq_{P,Cook} Y$ (or $X \leq_P Y$)

! Mind your direction, don't confuse $X \leq_\mathrm{P} Y$ with $Y \leq_\mathrm{P} \mathrm{X}$

Simplification: decision problems

- Search problem. Find some structure.
  - Example. Find a minimum cut.

- Decision problem.
  - Problem X is a set of strings [e.g., strings that encode graphs containing a triangle]
  - Instance: string s [e.g., encoding of a graph]
  - YES instance: $s \in X$; NO instance: $s \notin X$
  - Algorithm A solves problem X: $A(s) = yes$ iff. $s \in X$
  - Ex. Does there exist a cut of size $\leq k$?

Polynomial-time transformation

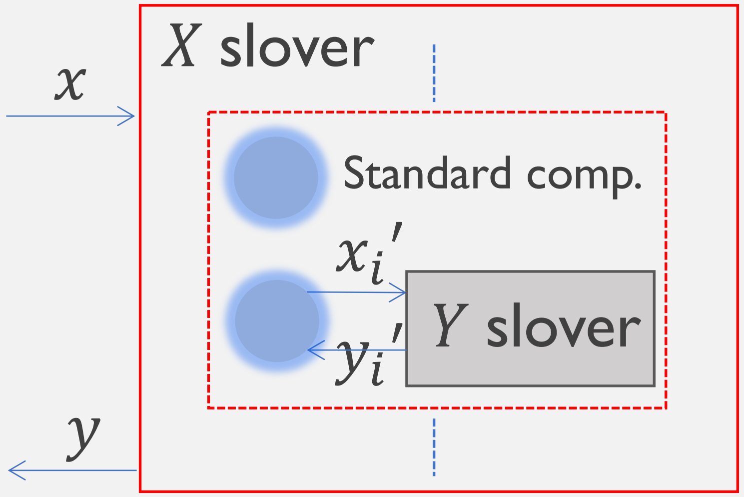
- Karp reduction. (Decision) problem X polynomial transforms to Problem Y if given any x, we can construct y such that
  - size $|y| = poly(|x|)$
  - $x \in X$ iff. $y \in Y$.

$$X \leq_{P,Karp} Y$$

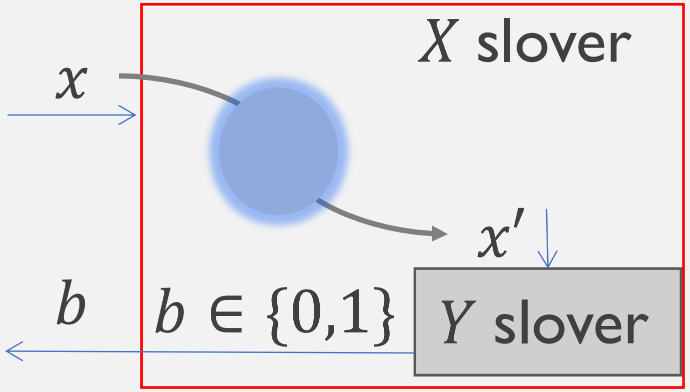
Polynomial-time reduction vs. transformation

Cook (Turing) reduction vs. Karp reduction

$X \leq_{P,Cook} Y$



$X \leq_{P,Karp} Y$ (Decision problems)



N.B. Polynomial transformation is polynomial reduction with just one call to oracle for Y, exactly at the end of the algorithm for X.

Open question. Are these two concepts equivalent?

Basic reduction strategies

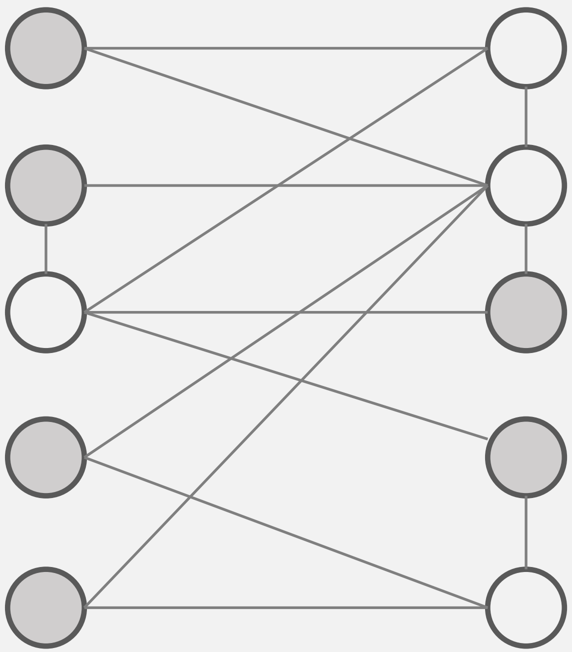
- Reduction by simple equivalence
- Reduction from special case to general case
- Reduction by encoding with gadgets

Independent set

Input. Graph $G = (V, E)$ and an integer k

- Independent set $S \subseteq V$: subset of vertices such that for each edge at most one of its endpoints is in S

Goal. Decide if there is an independent set S with $|S| \geq k$



 independent set

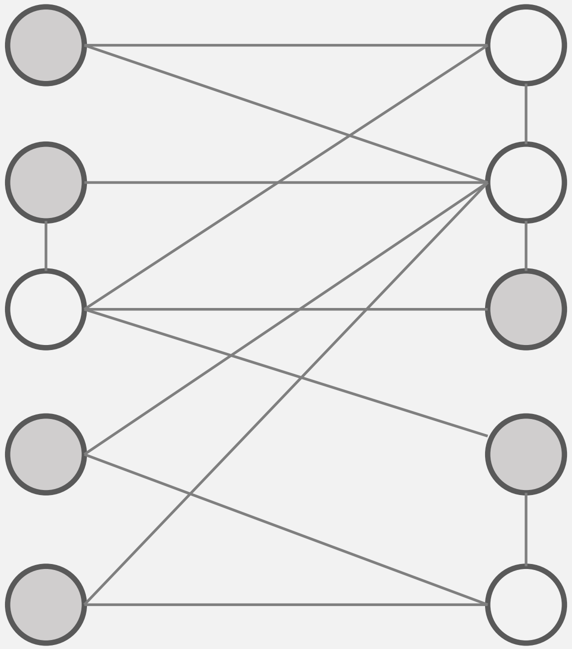
- Is there an independent set of size ≥ 6? ☺
- Is there an independent set of size ≥ 7? ☹

Vertex cover

Input. Graph $G = (V, E)$ and an integer k

- Vertex cover $S \subseteq V$: subset of vertices such that for each edge at least one of its endpoints is in S

Goal. Decide if there is an vertex cover S with $|S| \leq k$



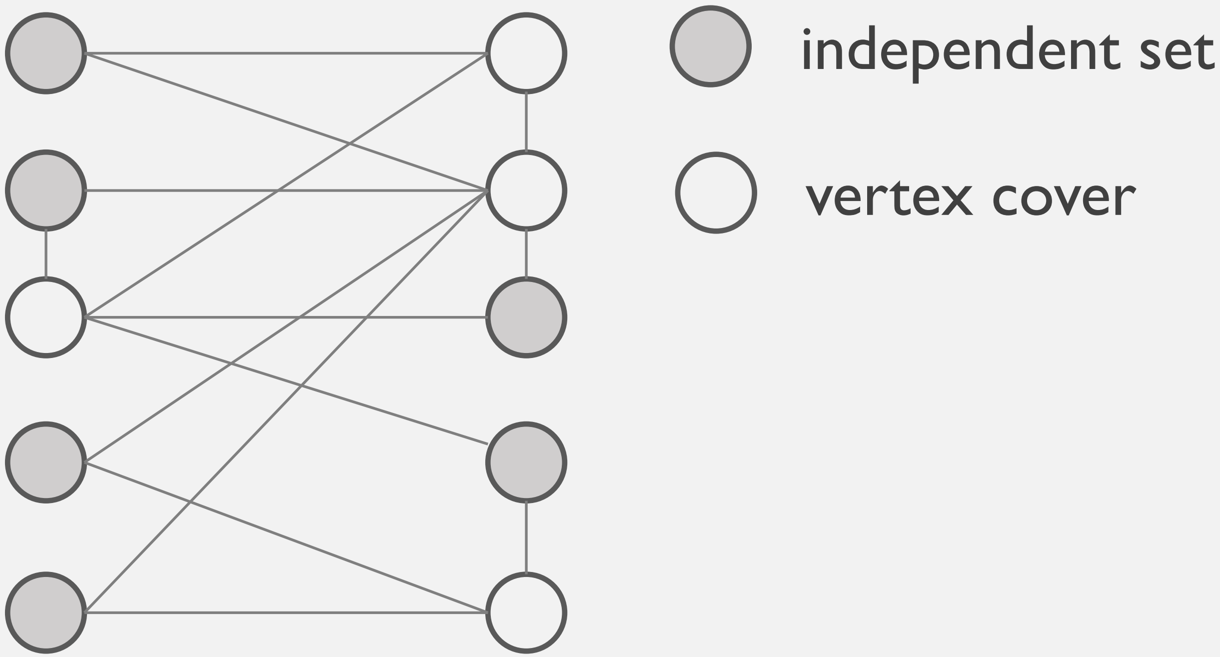
 Vertex cover

- Is there an vertex cover of size ≤ 4? ☺
- Is there an independent set of size ≤ 3? ☹

Independent set and Vertex cover

Claim. $\text{VERTEX}-\text{COVER} \equiv_P \text{INDEPENDENT}-\text{SET}$

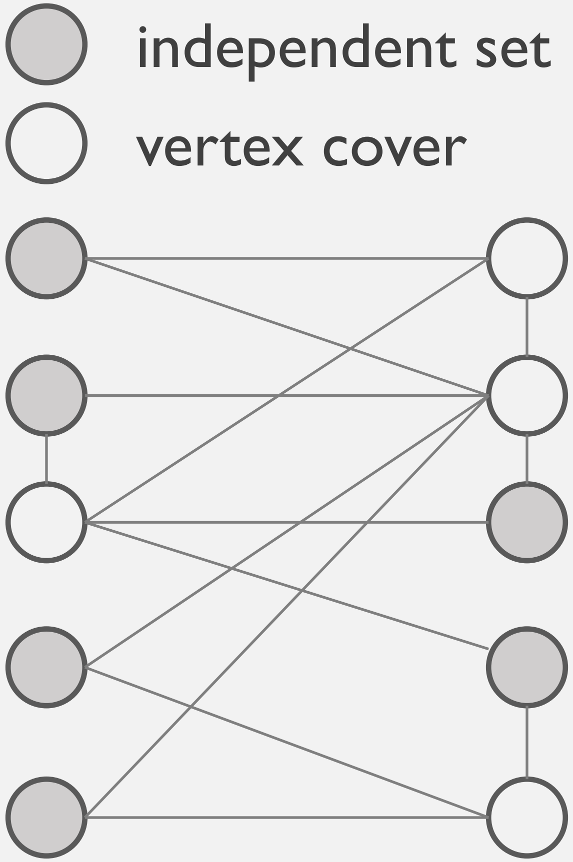
Pf. We show S is an independent set iff. $V \backslash S$ is a vertex cover



Independent set and Vertex cover

Claim. $\text{VERTEX}-\text{COVER} \equiv_P \text{INDEPENDENT}-\text{SET}$

Pf. We show S is an independent set iff. $V\backslash S$ is a vertex cover



$\leq$ ($\Leftarrow$) Let S be any independent set

- Consider an arbitrary edge (u, v)
- S independent $\Rightarrow u \notin S$ or $v \notin S \Rightarrow u \in V\backslash S$ or $v \in V\backslash S$
- Thus $V\backslash$S covers (u, v)

$\geq$ ($\Rightarrow$) Let $V\backslash S$ be any vertex cover

- Consider two nodes $u \in S$ and $v \in S$
- Observe that $(u, v) \notin E$ since $V\backslash S$ is a vertex cover
- Thus no two nodes in S are joined by an edge

$\Rightarrow$ S is an independent set

Basic reduction strategies

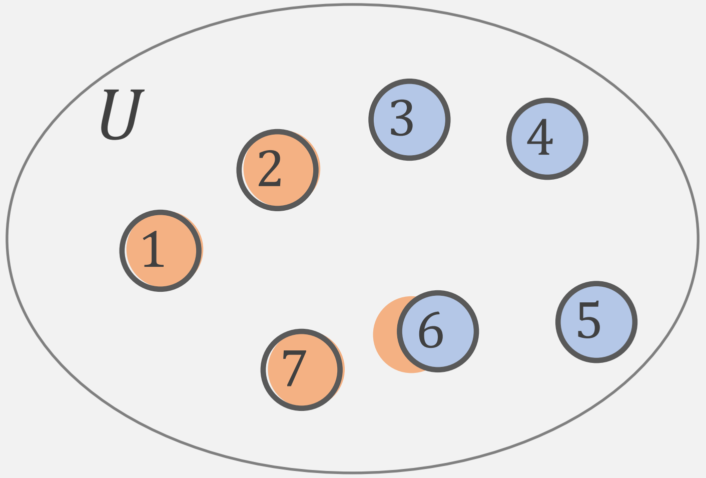
- Reduction by simple equivalence
  - $\text{VERTEX}-\text{COVER} \equiv_P \text{INDEPENDENT}-\text{SET}$
- Reduction from special case to general case
- Reduction by encoding with gadgets

Set cover

Input. Set U of n elements, $S_1, \dots, S_m$ of subsets of U, integer k

Goal. Decide if there is a collection of $\leq k$ of these sets whose union is equal to U

$U = \{1,2,3,4,5,6,7\}$
$k = 2$
$S_1 = \{3,7\}$, $S_2 = \{3,4,5,6\}$
$S_3 = \{1\}$, $S_4 = \{2,4\}$
$S_5 = \{5\}$, $S_6 = \{1,2,6,7\}$



Sample application.

- Set U of n capabilities that our computer system needs to have.
- m available pieces of software, ith software provides the set $S_i \subseteq U$ capabilities.
- Goal: achieve all n capabilities using fewest pieces of software.

Vertex cover reduces to set cover

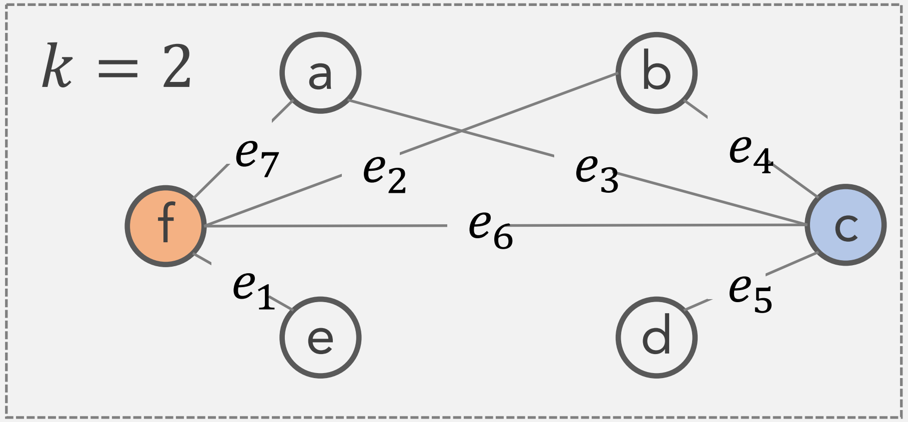
Claim. $\text{VERTEX}-\text{COVER} \leq_P \text{SET}-\text{COVER}$

Pf. Given a $\text{VERTEX}-\text{COVER}$ instance $G = \langle (V, E), k \rangle$, we construct a $\text{SET}-\text{COVER}$ instance whose solution size equals the size of the $\text{VERTEX}-\text{COVER}$ instance

Reduction: on input $\langle G = (V, E), k \rangle$

Output: // a SET-COVER instance

$k = k, U = E, S_v = \{e \in E : e \text{ incident to } v\}$ for every $v \in V$



$\Rightarrow$

$U = \{1,2,3,4,5,6,7\}$

$k = 2$

$S_a = \{3,7\}$, $S_c = \{3,4,5,6\}$

$S_e = \{1\}$, $S_b = \{2,4\}$

$S_d = \{5\}$, $S_f = \{1,2,6,7\}$

Basic reduction strategies

- Reduction by simple equivalence
  - $\mathrm{VERTEX-COVER} \equiv_P \mathrm{INDEPENDENT-SET}$
- Reduction from special case to general case
  - $\mathrm{VERTEX-COVER} \leq_P \mathrm{SET-COVER}$
- Reduction by encoding with gadgets

Satisfiability

- Literal: A Boolean variable or its negation x_i or $\overline{x_i}$
- Clause: A disjunction (OR) of literals $C_j = x_1 \vee \overline{x_2} \vee x_3$
- Conjunctive normal form: A propositional formula that is conjunction (AND) of clauses $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$

SAT. Given CNF formula Φ, is there a satisfying truth assignment?

EX. $(\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$
YES. $x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}$

3-SAT. SAT where each clause contains exactly 3 literals

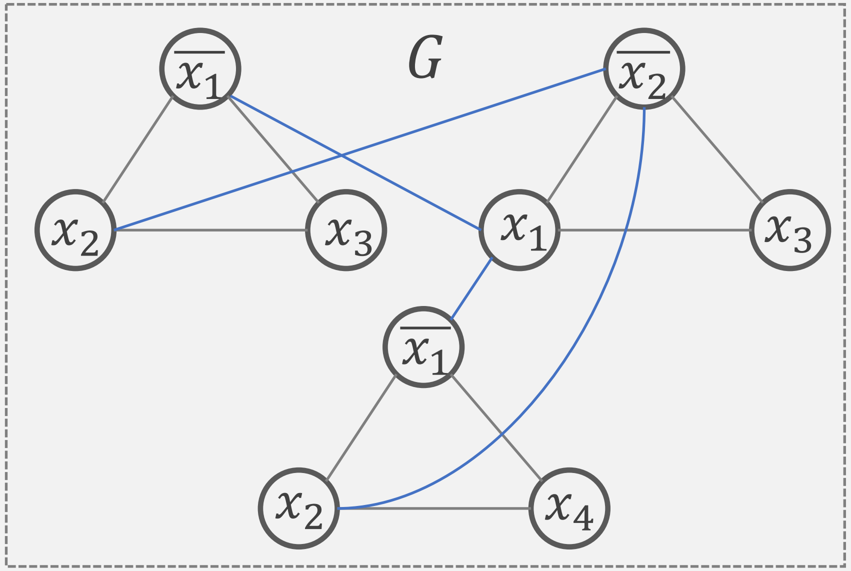
Reducing 3-SAT to independent set

Claim. $3\text{–SAT} \le_P \text{INDEPENDENT–SET}$

Pf. Given a 3–SAT instance Φ, we construct an INDEPENDENT–SET instance (G, k) that has an ind. set of size k iff. Φ is satisfiable.

Reduction: on input Φ

- Let G contain 3 vertices for each clause, one for each literal
- Connect 3 literals in a clause in a triangle
- Connect literal to each of its negations
- $k = |\Phi|$ \\ k=# clauses in Φ
- Output: $\langle G, k \rangle$



$\Uparrow$

$k = 3$ $\quad \Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)$

3-SAT reduces to independent set

Claim. $3\text{-SAT} \leq_P \text{INDEPENDENT-SET}$

Pf. Given a 3-SAT instance Φ, we construct an INDEPENDENT-SET instance $\boxed{(G, k) \text{ that has an ind. set of size } k \text{ iff. } \Phi \text{ is satisfiable.}}$

⇒ Let S be an independent set of size k

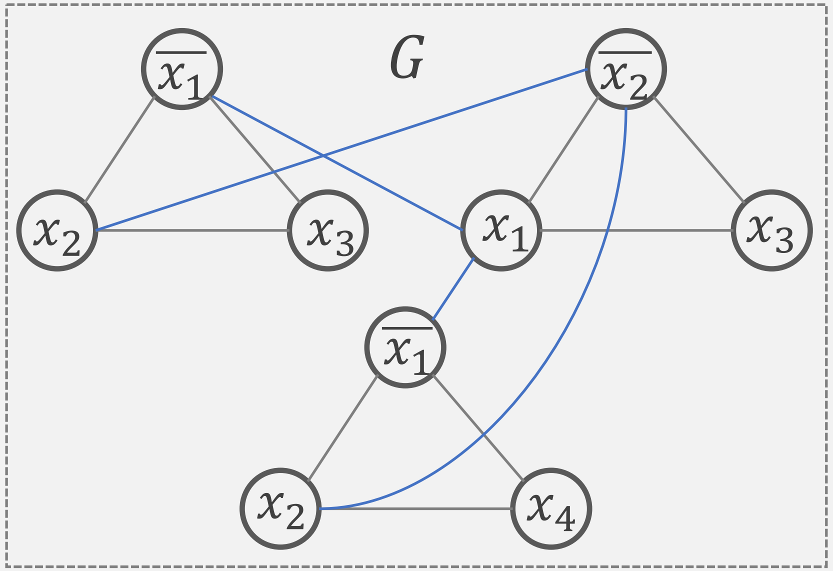
- S must contain exactly one vertex in each triangle
- Set these literals true (make others consistent)

→A valid assignment & all clauses satisfied

⇐ Given satisfying assignment

- Select one true literal from each triangle

→ An independent set of size k



Reflection on reductions

- Basic reduction strategies
  - Reduction by simple equivalence
  - Reduction from special case to general case
  - Reduction by encoding with gadgets

Transitivity. If $X \leq_P Y$ and $Y \leq_P Z$, then $X \leq_P Z$

Proof idea. Compose two reduction algorithms

➔ 3-SAT $\leq_P$ INDEPENDENT-SET $\leq_P$ VERTEX-COVER $\leq_P$ SET-COVER

Central ideas in complexity

- Poly-time as "feasible"
  - Most natural problems either are easy (e.g., n^3) or no poly-time alg. known
- Reduction : relating hardness ($A \leq B \Rightarrow A$ no harder than B)
- Classify problems by "hardness"

Self reducibility

Decision problem. Does there exist a vertex cover of size $\leq k$?
Search problem. Find vertex cover of minimum cardinality.

Self-reducibility. Search problem $\leq_P$ decision version

- Applies to all (NP-complete) problems in this chapter
- Justifies our focus on decision problems

Definition of class P

P. Decision problems for which there is a poly-time algorithm

Problem	Description	Algorithm	YES instance	No instance
Multiple	Is x a multiple of y?	Grade school	51,17	52,17
RELPRIME	Are x and y relatively prime?	Euclid (300 BCE)	34,39	34,51
PRIMES	Is x a prime?	AKS 2002	53	51
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	neither either	algorithm quantum

Definition of class NP

NP. Decision problems for which there is a poly-time certifier

Idea of certifier

- Certifier checks a proposed proof π that $s \in X$
- Need not determine whether $s \in X$ on its own

N.B. $|t| = p(|s|)$ for some polynomial $p()$

Def. Algorithm $C(s,t)$ is a certifier for problem X if for every string s, $s \in X$ iff there exists a string t such that $C(s,t) = \text{yes}$

Equivalent def. $\mathbf{NP}$ = nondeterministic polynomial−time

not ~~p~~olynomial−time

Certifiers and certificates: Composite

COMPOSITES. Given an integer s, is s composite?

- Certificate: A non-trivial factor t of s.
- Certifier.
- Instance. $s = 437{,}669$
  - Certificate. $t = 541$ *or* 809. $437{,}669 = 541 \times 809$

```
CompositesCertifier(s,t)
 If (t ≤ 1 or t ≥ s)
      Return false
 Else if (s is a multiple of t)
      Return true
 Else
      Return false
```

Conclusion. COMPOSITES $\in$ **NP**

Certifiers and certificates: Hamiltonian cycle

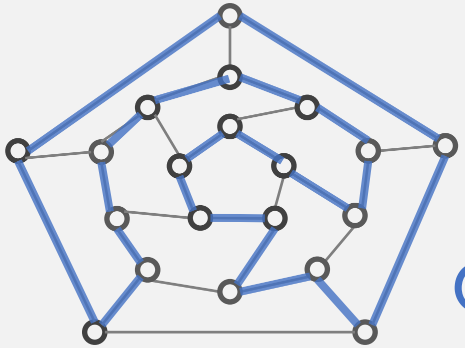
HAM−CYCLE. Given an undirected graph $G = (V, E)$, does there exist a simple cycle that visits every node?

- Certificate: A permutation of n nodes
- Certifier.

```
HAM-CYCLE-Certifier(G, σ)
  If (∀i, j, σ_i ≠ σ_j &(σ_i, σ_{i+1}) ∈ E)
     Return true
```

Conclusion. HAM−Cycle ∈ **NP**

Instance s



Certificate t

P,NP,EXP

P. Decision problems for which there is a poly-time algorithm

EXP. Decision problems for which ∃ an exponential-time algorithm

i.e., runs in time $O(2^{p(|s|)})$ for some polynomial $p()$

NP. Decision problems for which there is a poly-time certifier

- Claim. $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$

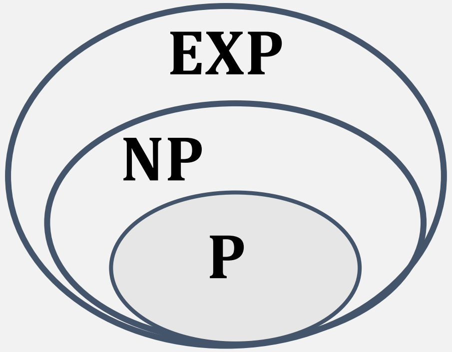
$\mathbf{P} \subseteq \mathbf{NP}$. Consider any $X \in \mathrm{P}$,
- ∃ poly−time A that solves X
- Certificate: $t = \epsilon$, certifier $C(s,t) = A(s)$

$\mathbf{NP} \subseteq \mathbf{EXP}$. Consider any $X \in \mathrm{NP}$,
- ∃ poly−time certifier $C(s,t)$
- To decide input s, run $C(s,t)$ on all strings t with $|t| \leq p(|s|)$.
- Return yes, if $C(s,t)$ ever says yes.

Open question: P = NP?

The Millennium prize problems

- $1 million prize



- Consensus opinion on P = NP? Probably no.

Eight Signs A Claimed P≠NP Proof Is Wrong

As of this writing, Vinay Deolalikar still hasn't retracted his P≠NP c

https://www.scottaaronson.com/blog/?p=458

Millennium Problems

Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the
no proof of this property is known.

Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. Th
average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious'

P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve th
the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, ho
solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation
This is the equation which governs the flow of fluids such as water and air. Howeve
solutions exist, and are they unique? Why ask for a proof? Because a proof gives no

Hodge Conjecture
The answer to this conjecture determines how much of the topology of the solutio
further algebraic equations. The Hodge conjecture is known in certain special case
dimension four it is unknown.

Poincaré Conjecture
In 1904 the French mathematician Henri Poincaré asked if the three dimensional s
manifold. This question, the Poincaré conjecture, was a special case of Thurston's g
three manifold is built from a set of standard pieces, each with one of eight well-ur

Birch and Swinnerton-Dyer Conjecture