Last Time

- Graph Algorithms
Today

- NP-Completeness
Hamiltonian Graph

- **Input**: G
- **Problem**: Does G contain a cycle that goes through every node exactly once – a Hamiltonian Cycle?
Overview of NP-Completeness

- Define P – Polynomial time **decidable** algorithms.
- Define NP – Polynomial time **verifiable** algorithms.
- Examples of NP-Complete problems.
- Define NP-Completeness.
- Reductions to show NP-Completeness.
Definition of P

Any problem \( p \in P \) can be **solved** for all inputs of size \( n \) in \( O(n^k) \) time.

Examples:

- Sort \( n \) elements
- Find the shortest path between two nodes in a graph with \( n \) nodes
- Generate a Huffman Tree
- ...Everything we’ve done in the class so far.
Any problem $\pi \in NP$ can be verified for all inputs of size $n$ in $O(n^k)$.

Given a “certificate” $y$, can we verify that the certificate is correct in polynomial time?
A problem is NP if there exists a poly-time algorithm $V(x, y)$ s.t. such that

- $x$ is a \textbf{YES} instance of $\pi$ if there exists $y$, $|y| \leq poly(|x|)$ s.t. $V(x, y) = \text{YES}$
- $x$ is a \textbf{NO} instance of $\pi$ if for all $y$, $|y| \leq poly(|x|)$ s.t. $V(x, y) = \text{NO}$

- Short certificates
- $V(x, y)$ is a poly-time algorithm
Verifier V

Input: instance of $\pi$

Verification: $y$ - certificate

Output: {Yes, No}
Examples of NP problems

Node Cover

- **Input:** $G, k$
- **Problem:** Does there exist a node-cover of $G$ with $\leq k$ nodes?
- **Certificate:** A set of $k$ nodes.
Examples of NP problems

Hamiltonian Graph

- **Input**: G
- **Problem**: Does G contain a Hamiltonian Cycle? (A cycle that goes through every node exactly once.)
- **Certificate**: A permutation of the $V(G)$. 
Examples of NP problems

Traveling Salesman

- **Input**: G with weights w.
- **Optimization Problem**: Find a minimal hamiltonian cycle.
- **Decision Problem**: Does there exist a cycle with weight < k?
- **Certificate**: A permutation of the V(G).
Examples of NP problems

Composite Number

- **Input**: An integer $N$
- **Problem**: Is $N$ composite?
- **Certificate**: A factor of $N$. 
Examples of NP problems

Satisfiability of a boolean formula

- **Input**: A boolean formula $\phi$ e.g. $(x_1 \lor x_3) \land (x_2 \lor (x_3 \land \neg x_1))$
- **Problem**: Is there an assignment of variables s.t. $\phi$ is $\text{TRUE}$? Is $\phi$ satisfiable?
- **Certificate**: A truth assignment for each variable.
$P \subseteq NP$

Since we can solve any problem in $P$ in poly-time, we can certainly verify a solution in poly-time.
Assume we have two decision problems, $A$ and $B$. Find a poly-time algorithm, $f$, that maps instances of $A$ to instances of $B$.

- **YES** instances of $A$ $\rightarrow$ **YES** instances of $B$.
- **NO** instances of $A$ $\rightarrow$ **NO** instances of $B$.

Then, $A \leq_p B$. “$A$ can be solved in at most the amount of time to solve $B$.”

- If $B \subseteq P$ then $A \subseteq P$
- If $A \not\subseteq P$ then $B \not\subseteq P$
- If $B \subseteq NP$ then $A \subseteq NP$

Also, $\leq_p$ is transitive. $A \leq_p B$ and $B \leq_p C$ then $A \leq_p C$. 
Define NP-Completeness

$B$ is NP-complete (NPC) if:

- $B \in NP$
- $B$ is NP-hard.

$B$ is NP-hard if:

- For every problem $A \in NP$, $A \leq_p B$

**Claim:** Either all NP-complete problems $\in P$ or None are.

**Proof:** $B \in NPC$. If $B \in P$ then $A \in P$ for all $A \in NPC$. 

Relationship between P and NP
Method for Proving NP-Completeness

1. Prove $A \in NP$
   - Show that a certificate can be verified in poly-time.

2. Select a known NP-Complete problem, $B$.

3. Describe an algorithm $f$ that maps every instance $x$ of $B$ onto an instance $f(x)$ of $A$

4. Prove that $x$ satisfies $B$ if and only if $f(x)$ satisfies $A$.

5. Prove that $f$ runs in polynomial time.
Example reduction:

Take as given that $\text{Clique}$ is NPC.

$\text{Clique} = \{ (G, k) : G \text{ contains a clique of size } k \}$

A clique is a set of nodes, such that there is an edge between every pair of nodes in the clique.
**Vertex-cover**

\[ \text{Vertex-Cover} = \{ (G, k) : G \text{ has a vertex-cover of size } k \} \]

A **vertex cover** is a set of nodes, such that every edge in the graph is adjacent to a node in the vertex cover.

**Claim:** \text{Vertex-Cover} is NP-Complete.
1. Show \textsc{Vertex-Cover} \(\in\) \textit{NP}

Choose a set of vertices \(V' \subseteq V\) as a certificate. Verifying \textsc{Vertex-Cover} consists of inspecting every edge \((u, v) \in E\) and checking if \(u \in V'\) or \(v \in V'\). This is \(O(V'E)\) – polynomial.
2. Show \text{Clique} \leq_p \text{Vertex-Cover}.

Since \text{Clique} is NP-hard, then this will show that \text{Vertex-Cover} is NP-hard, and thus, NP-Complete.

Given an undirected graph $G = (V, E)$, let the \textbf{complement} of $G$ be $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{(u, v) : u, v \in V$ and $(u, v) \notin E\}$.

\textbf{Claim}: $G$ has a clique of size $k$ iff $\bar{G}$ has a vertex cover of size $|V| - k$. 
Example

G

\[ G \]

G

\[ G \]
Claim: $G$ has a clique of size $k$ iff $\bar{G}$ has a vertex cover of size $|V| - k$.

Proof: Suppose $G$ has a clique $V' \subseteq V$ and $|V'| = k$. Claim that $V - V'$ is a vertex cover of $\bar{G}$.

- Let $(u, v) \in \bar{E}$.
- $(u, v) \notin E$. So, one of $u$ and $v$ is not in $V'$, since every pair of nodes in $V'$ are connected by an edge.
- Thus, one of $u$ and $v$ is in $V - V'$
- Therefore the edge $(u,v)$ is covered by a vertex in $V - V'$.
- Since this holds for all edges in $\bar{E}$, $V - V'$ is a vertex cover of $\bar{G}$ with size $= |V| - k$
**Claim**: $G$ has a clique of size $k$ iff $\bar{G}$ has a vertex cover of size $|V| - k$.

**Proof**: Suppose $\bar{G}$ has a vertex cover $V' \subseteq V$ and $|V'| = |V| - k$. Claim that $V - V'$ is a clique in $G$.

- For all $u, v \in V$, if $(u, v) \in \bar{E}$ then $u \in V'$ or $v \in V'$ or both – by defn of vertex cover.

- Thus, for all $u, v \in V$ if $u \notin V'$ and $v \notin V'$, then $(u, v) \notin \bar{E}$ so $(u, v) \in E$.

- So, for every pair of vertices $u, v \in V - V'$, $(u, v) \in E$, thus $V - V'$ is a clique.

- $|V - V'| = |V| - k$
Method for Proving \textsc{Vertex-Cover} is NP-Complete

1. Prove \textsc{Vertex-Cover} $\in$ \textsc{NP}
   - A set of vertices can be shown to be a vertex cover in poly time

2. Select a known NP-Complete problem, \(B\).
   - \(B = \text{Clique}\)

3. Describe an algorithm \(f\) that maps every instance \(x\) of \(B\) onto an instance \(f(x)\) of \(A\)
   - Construct \(\tilde{G}\)

4. Prove that \(x\) satisfies \(B\) if and only if \(f(x)\) satisfies \(A\).
   - Previous proofs.

5. Prove that \(f\) runs in polynomial time.
   - Constructing \(\tilde{G}\) takes \(O(V + E)\).
NP-complete problems that may or may not be solvable in polynomial time.

If NP-complete problem can be solved in poly time, they all can.

NP-completeness proofs rely on showing the equivalence (in poly time) between a new problem and a known NP-complete problem.

**Logical gap:** the initial NP-complete problem: **Circuit-Sat**.
- Cormen has an outline of the proof.
Bye

- Next time
  - Hashing