SCS1308 Foundations of Algorithms

Tutorial Session - 06

How to prove problem is NP, NP-Complete and NP-Hard

Introduction to NP Problems

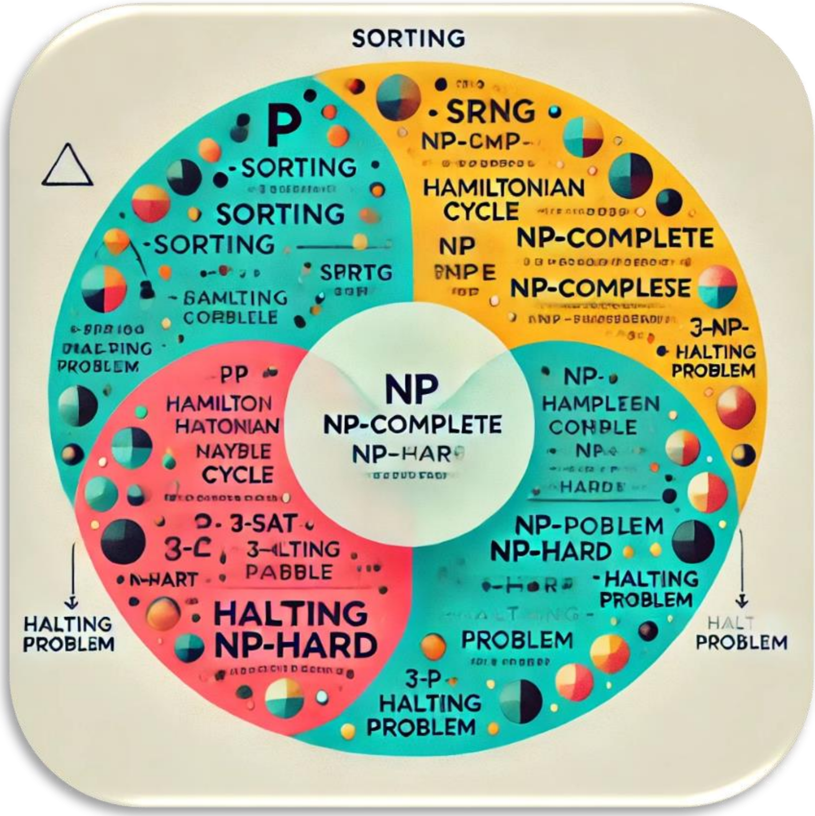
Imagine a university with:

P Problems: These are easy exams that anyone can finish on time (e.g., sorting).

NP Problems: These are tricky exams. Grading answers is easy, but solving them is hard (e.g., 3-SAT).

NP-Complete Problems: The hardest exams in the university. If someone finds a quick way to solve one, they've solved all NP exams.

NP-Hard Problems: Impossible exams. Even if you are given the solution, you might not be able to verify it on time (e.g., optimization problems).



Introduction to NP Problems

Problem Type	Definition	Examples
P	Solvable in polynomial time.	Sorting, Shortest Path.
NP	Verifiable in polynomial time.	Subset Sum, 3-SAT.
NP-Complete	Hardest problems in NP. Solving one efficiently solves all NP problems.	TSP (Decision), 3-SAT.
NP-Hard	As hard as NP-Complete. May not be verifiable in polynomial time.	TSP (Optimization), Halting Problem.

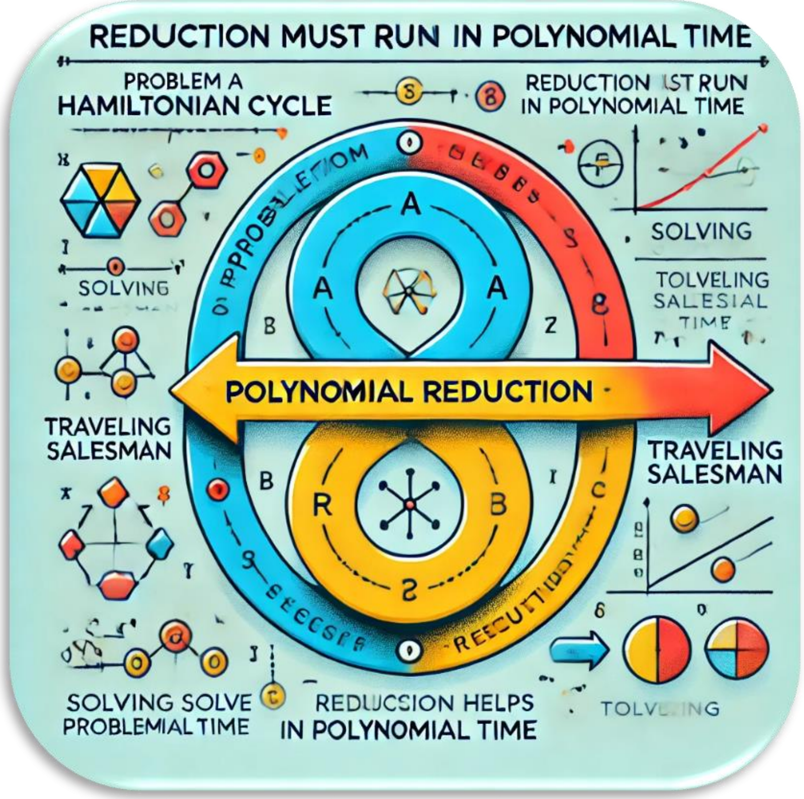
Polynomial Reductions: Idea

“Reductions are a common and powerful concept in computer science. The basic idea is that we solve a new problem by reducing it to a known problem.”

In complexity theory we want to use reductions that allow us to prove statements of the following kind:

- **Problem A can be solved efficiently**
- **if problem B can be solved efficiently.**

For this, we need a reduction from **A** to **B** that can be computed efficiently itself (otherwise it would be useless for efficiently solving A).



Polynomial Reductions: Definition

Let **A** $\subseteq \Sigma *$ and **B** $\subseteq \Gamma *$ be decision problems. We say that **A** can be polynomial reduced to **B**, written A $\leq$p B, if there is a function **f** : $\Sigma * \rightarrow \Gamma *$ such that:

- f can be computed in polynomial time by a **DTM**
  - i. e., there is a polynomial **p** and a DTM **M** such that M computes f (w) in at most p(|w|) steps given input w $\in \Sigma *$
- f reduces A to B
  - i. e., for all w $\in \Sigma *$: w $\in$ A iff f (w) $\in$ B f is called a polynomial reduction from A to B

f is called a polynomial reduction from A to B

Polynomial Reductions: Example (1)

Hamilton Cycle is the following decision problem:

- Given: undirected graph $G = (V, E)$
- Question: Does G contain a Hamilton cycle?

A Hamilton cycle of G is a sequence of vertices in V,

- $\pi = \mathbf{v}_0, \ldots, \mathbf{v}_n$, with the following properties:
- π is a path: there is an edge from vi to vi+1 for all $0 \leq i < n$
- π is a cycle: $\mathbf{v}_0 = \mathbf{v}_n$
- π is simple: $\mathbf{v}_i \neq \mathbf{v}_j$ for all $i \neq j$ with $i, j < n$

π is Hamiltonian: all nodes of V are included in π

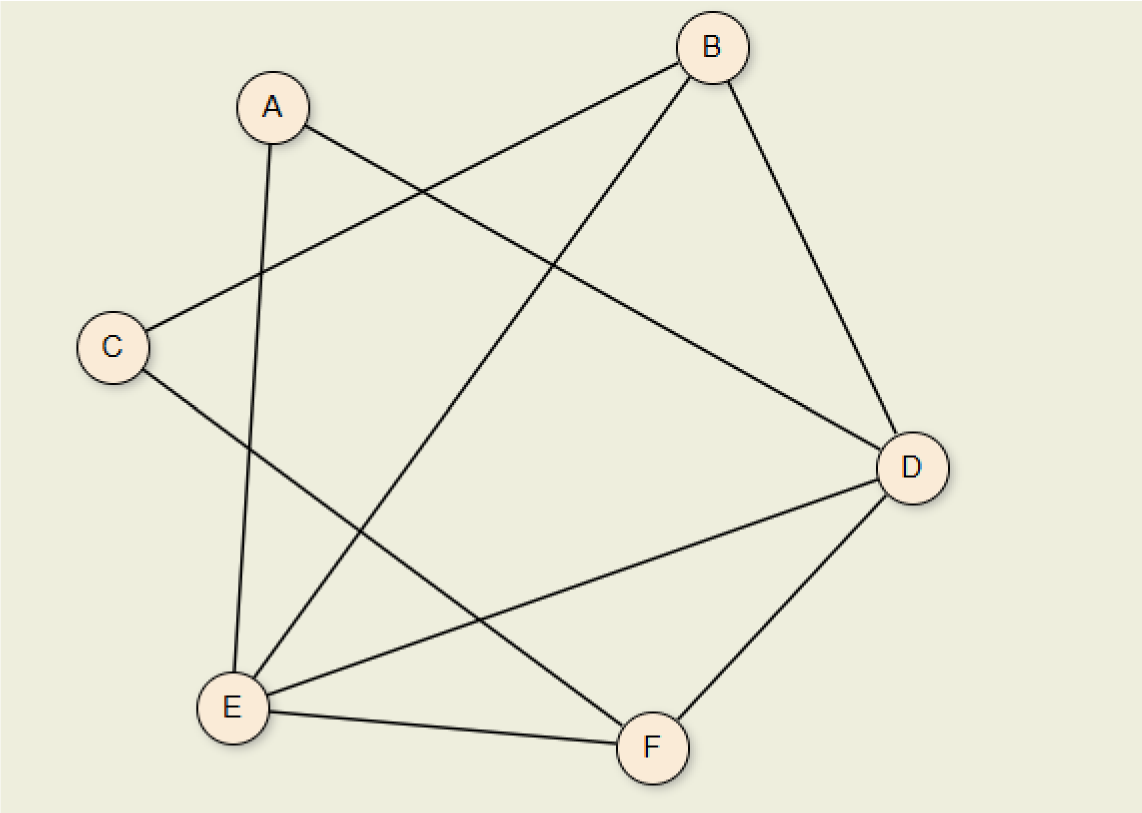
Polynomial Reductions: Example (2)

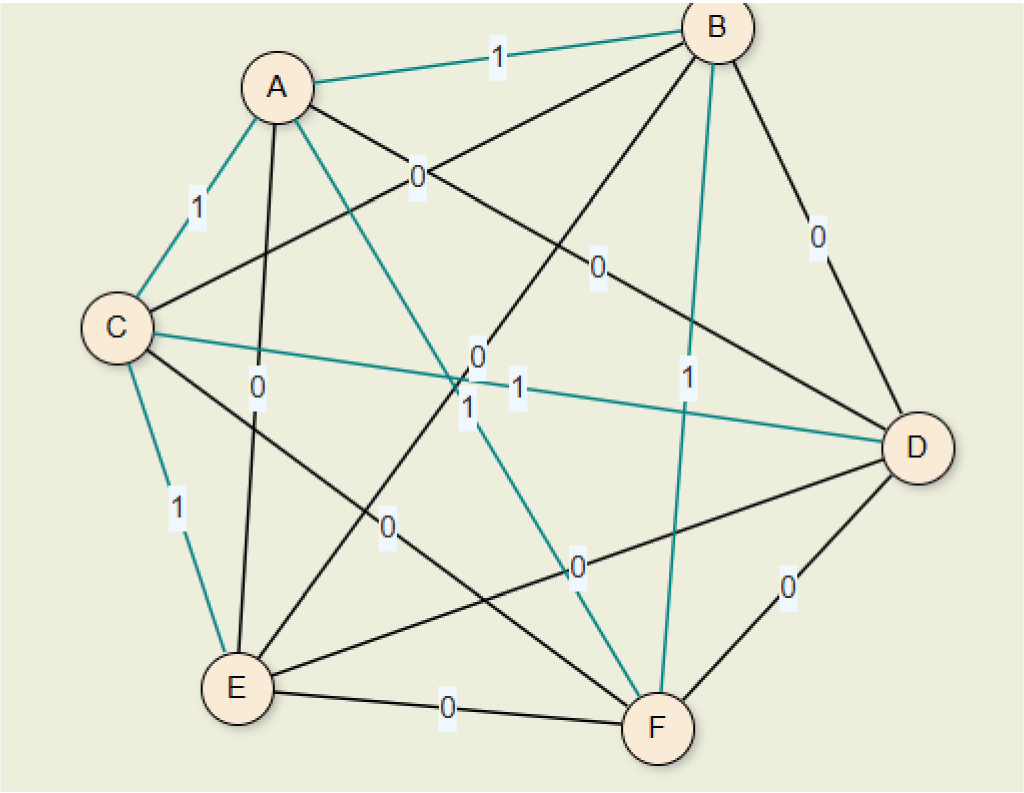
For a given graph G (V, E), the **HAMILTONIAN CYCLE** problem is to find whether G contains a Hamiltonian Cycle that is, a cycle that passes through all the vertices of the graph **exactly once**.

For a given weighted graph G' = (V', E'), with **non-negative weights**, and integer k', the **TRAVELING SALESMAN** problem is to find whether G' contains a simple cycle of length < k that passes through all the vertices. [The length of a cycle is the sum of weights of all the edges in the cycle.]

Polynomial Reductions: Example (2)

Let graph G be an input to HAMILTONIAN CYCLE





The constructed graph G' . The blue edges were not present in G and so have weight of 1.

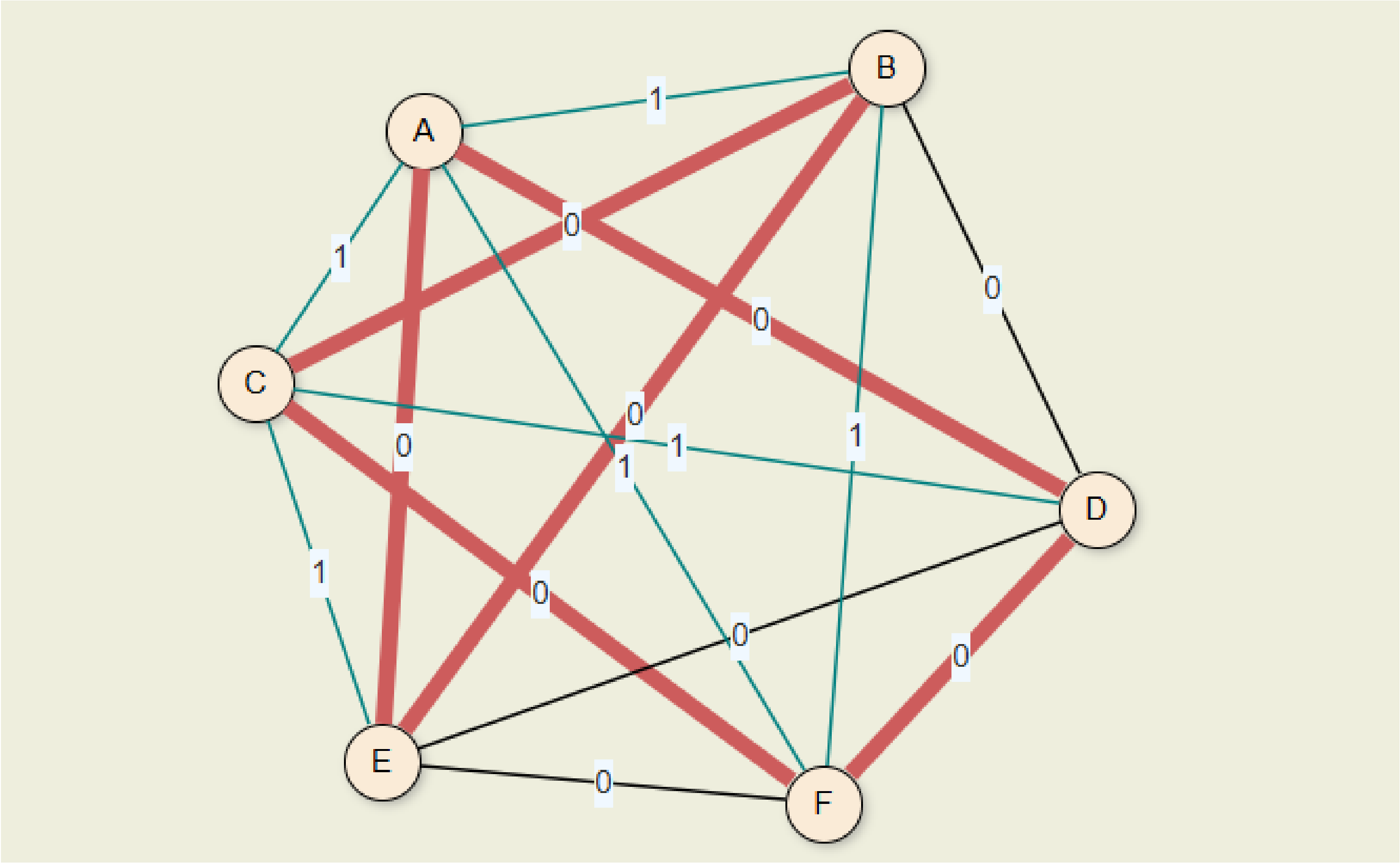
Reduction of HAMILTONIAN CYCLE to TSP

- The graph G has a Hamiltonian Cycle if and only if there exists a cycle in G' passing through all vertices exactly once, and that has a length = 0 (i.e., has a solution for the instance of **TRAVELING SALESMAN** where k = 0.

- If there is a cycle that passes through all vertices exactly once, and has length = 0 in G',
  - the cycle contains only edges that were originally present in graph G. (The new edges in G' have weight 1 and hence cannot be part of a cycle of length = 0.) Hence there exists a **Hamiltonian cycle** in G.
  - If there exists a Hamiltonian Cycle in G, it forms a cycle in G' with length = 0, since the weights of all the edges is 0. Hence there exists a solution for **TRAVELING SALESMAN** in G' with length = 0.

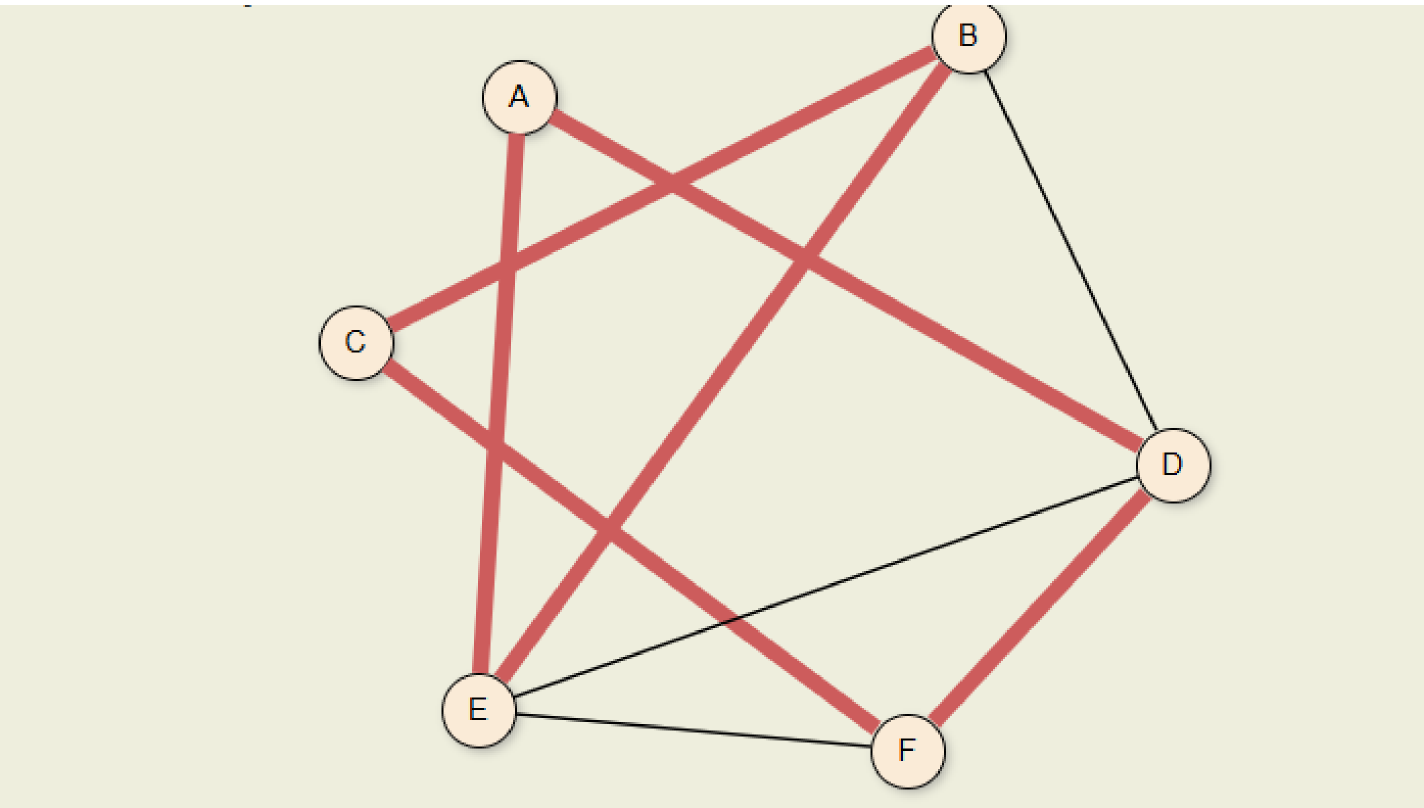
Reduction of HAMILTONIAN CYCLE to TSP

G' has a cycle passing through all vertices exactly once with length = 0.



Reduction of HAMILTONIAN CYCLE to TSP

G' has a cycle passing through all vertices exactly once with length = 0. This is a **Hamiltonian cycle** in G.



Activity 01:

Question:
The Traveling Salesman Problem (TSP) is a well-known optimization problem, while the Hamiltonian Cycle problem (HCP) is a graph-based decision problem. Using the following steps, describe how the Hamiltonian Cycle problem can be reduced to the Traveling Salesman Problem.

1. Understanding the Problems:

- **Hamiltonian Cycle Problem (HCP)**:
  Input: An unweighted graph G=(V,E)
  Output: Determine whether there exists a cycle that visits each vertex exactly once.
- **Traveling Salesman Problem (TSP)**:
  Input: A weighted graph G’=(V,E,w) and a budget B.
  Output: Determine whether there exists a tour visiting all vertices with a total weight ≤B.

2. Why does this reduction prove that TSP is NP-Hard?
3. How does it establish that solving TSP in polynomial time would solve HCP?