Algorithms, Design & Analysis

Lecture 27: Graph Problems, Reductions, and NP-Completeness

Afrazia Umer Farooq & Qudsia Khan

Information Technology University

May 8, 2025

About Your Fellows

- Hello, our name is **Afrazia Umer Farooq** and **Qudsia Khan**. .
- We are Associate Students at ITU.

Overview

- Recap: Key Concepts from Previous Lectures
- Understanding NP and Verification
- Hamiltonian Path Problem: Definition and Examples
- Hamiltonian Cycle Problem: Definition and NP Membership
- Reduction: Hamiltonian Path to Hamiltonian Cycle with Complete Proof
- Longest Path Problem: Definition and Reduction from Hamiltonian Path
- Traveling Salesperson Problem (TSP): Definition and Variants
- Reduction: Hamiltonian Cycle to TSP
- Exercises and Key Takeaways
- Next Steps and Resources

Recap: Previous Concepts

- **Maximum Independent Set (MIS) to Minimum Vertex Cover (MVC):**
  - Reduced MIS to MVC to illustrate their complexity equivalence.
- **3-SAT to 4-Colorability:**
  - Utilized OR gadgets to reduce 3-SAT to 4-Colorability, proving NP-hardness.

Understanding NP – Definition

What is NP?

A problem belongs to NP if a "yes" instance can be verified in polynomial time using a certificate.

- Example: Verifying a Hamiltonian Cycle in a graph.

Understanding NP – Two-Party Model

Two-Party Model for Verification

A framework to understand NP through interaction between two entities.

- **Challenger:** Provides the problem instance (e.g., a graph).
- **Prover:** Supplies a certificate (e.g., a vertex list) claiming a "yes" answer.
- **Verification:** The Challenger validates the certificate in polynomial time.

Recap – Composite Numbers Verification

Problem: Is X Composite?

Determine if integer X is not prime.

- **Certificate:** An integer a that divides X.
- **Verification:** Confirm X/a is an integer.
- **Time Complexity:** Polynomial.

Composite Numbers – Example

- **Example:** $X = 15$, certificate $a = 3$.
- **Verification:** $15/3 = 5$ (integer), confirming X is composite.

$$X = 15 \quad \div \quad a = 3 \quad = \quad 5$$

Hamiltonian Path Problem

(Finding a Path Visiting All Vertices Exactly Once)

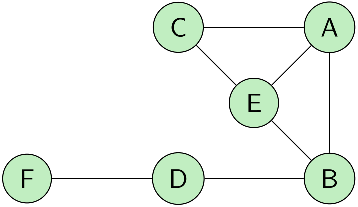
Hamiltonian Path Problem – Definition

Definition

Given a graph $G = (V, E)$ with $|V| = n$, find a simple path (no repeated vertices) that visits all vertices exactly once. The path length must be $n - 1$.

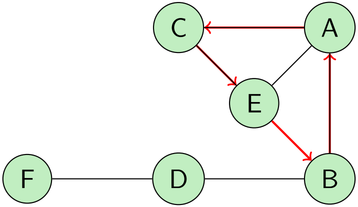
Hamiltonian Path Problem – Example Graph

- **Graph:** Vertices $\{A, B, C, D, E, F\}$.
- **Edges:** $F - D, D - B, B - A, A - C, C - E, E - B, A - E$.



Hamiltonian Path Problem – Failed Attempt

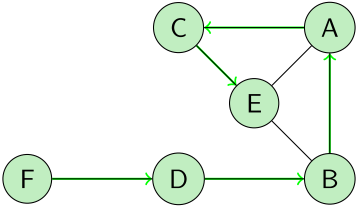
- **Attempt 1 (Failed):** $E \to B \to A \to C \to E$.
- **Reason:** Forms a cycle and misses vertices D and F.



Fails: Cycle, misses D, F

Hamiltonian Path Problem – Successful Attempt

- **Attempt 2 (Success):** $F \to D \to B \to A \to C \to E$.
- **Reason:** Visits all 6 vertices exactly once without forming a cycle.



Success: Visits all vertices

Hamiltonian Cycle Problem

(Finding a Cycle Visiting All Vertices Exactly Once)

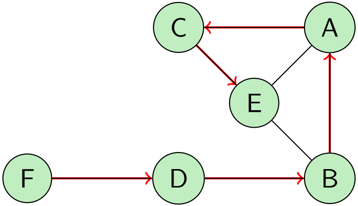
Hamiltonian Cycle Problem – Definition

Definition

Given a graph $G = (V, E)$ with $|V| = n$, find a cycle that visits every vertex exactly once and returns to the starting vertex. The cycle length must be n.

Hamiltonian Cycle Problem – Failed Attempt

- **Graph:** Vertices $\{A, B, C, D, E, F\}$.
- **Attempt (Failed):** $F \to D \to B \to A \to C \to E$.
- **Reason:** No edge from E to F, preventing cycle formation.



Fails: No edge $E \to F$

Hamiltonian Cycle Problem – Conclusion

- **Conclusion:** No Hamiltonian cycle exists in this graph.
- **Reason:** Vertex F lacks sufficient incoming edges to form a cycle.

Hamiltonian Cycle Problem – Proving NP-Hardness

Goal

Establish that the Hamiltonian Cycle problem is NP-Hard.

- **Approach:** Reduce a known NP-Hard problem, such as 3-SAT, to Hamiltonian Cycle.
- **Reduction Overview:** Construct a graph where a Hamiltonian cycle exists if and only if the 3-SAT instance is satisfiable.
- **Note:** This reduction will be explored in future lectures.

Reduction: Hamiltonian Path to Hamiltonian Cycle

(Proving NP-Hardness of Hamiltonian Cycle)

Reduction – Hamiltonian Path to Hamiltonian Cycle

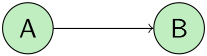
Objective

Reduce the Hamiltonian Path problem to the Hamiltonian Cycle problem to prove the latter's NP-Hardness.

- **Input:** Graph G with a potential Hamiltonian Path.
- **Output:** Graph H with a Hamiltonian Cycle if G has a Hamiltonian Path.
- **Construction:**
  - Copy graph G to create H.
  - Introduce a new vertex Z.
  - Add two-way edges between Z and all vertices in G.

Example Graphs – G_1 (Yes Instance)

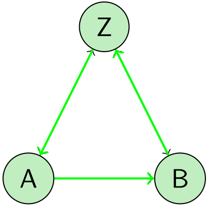
- **Graph** G_1**:** Contains a Hamiltonian Path.
- **Path:** $A \to B$.



Hamiltonian Path: $A \to B$

Example Graphs – H_1 (Yes Instance)

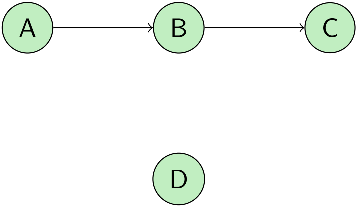
- **Graph H_1:** Add vertex Z, connect to all vertices.
- **Cycle:** $A \to B \to Z \to A$.



Hamiltonian Cycle: $A \to B \to Z \to A$

Example Graph – G_2 (No Hamiltonian Path)

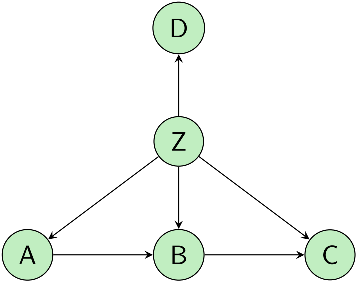
- **Graph G_2:** Directed acyclic graph with 4 nodes.
- **Reason:** No path can visit all vertices exactly once due to isolated node.



No Hamiltonian Path (D is unreachable)

Example Graph – H_2 (No Hamiltonian Cycle)

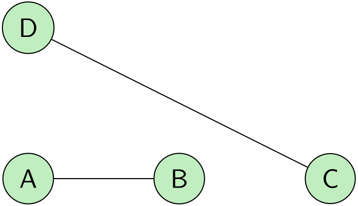
- **Graph H_2:** Extend G_2 by adding node Z, connect it to all others.
- **Reason:** Z is a source (no incoming edge), making a return impossible — cycle breaks.



No Hamiltonian Cycle: no return to Z

Example Graph – G_3 (No Hamiltonian Path)

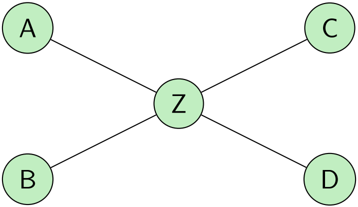
- **Graph G_3:** Undirected, disconnected components.
- **Reason:** No path covers all vertices due to disconnection.



No Hamiltonian Path (graph disconnected)

Example Graph – H_3 (No Hamiltonian Cycle)

- **Graph H_3:** Central node Z connects all, but still no valid cycle.
- **Reason:** Star structure prevents a cycle visiting all nodes once.



No Hamiltonian Cycle (star graph)

Proof of Reduction – Forward Direction

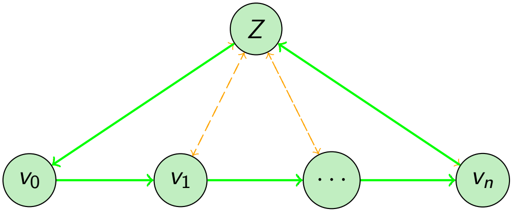
If G Has a Hamiltonian Path

Then H has a Hamiltonian Cycle.

- **Start with a Hamiltonian Path in G:** Consider a path $v_0 \to v_1 \to \cdots \to v_n$, visiting all vertices exactly once.
- **Build a Hamiltonian Cycle in H:**
  - In H, use the path and add two new edges: $v_n \to Z$ and $Z \to v_0$, where Z is a new auxiliary vertex.
  - This forms a cycle: $v_0 \to v_1 \to \cdots \to v_n \to Z \to v_0$.
- **Why it works:** The edges $v_n \to Z$ and $Z \to v_0$ exist in H, completing the cycle.

Conclusion: A Hamiltonian Path in G can be transformed into a Hamiltonian Cycle in H.

Proof of Reduction – Forward Direction (Visualization)



Cycle in H: $v_0 \to v_1 \to \cdots \to v_n \to Z \to v_0$

Proof of Reduction – Reverse Direction (Using Contraposition)

Objective

Prove that if H has a Hamiltonian Cycle, then G has a Hamiltonian Path.

- **Approach:** Use contraposition to simplify the proof.
- **Original Statement:** If G has no Hamiltonian Path ($\neg P$), then H has no Hamiltonian Cycle ($\neg Q$).
- **Contrapositive:** If H has a Hamiltonian Cycle (Q), then G has a Hamiltonian Path (P).

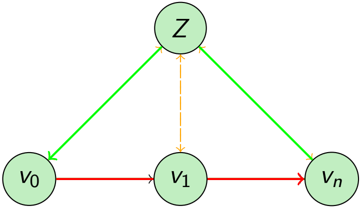
Proof of Reduction – Reverse Direction (Step-by-Step)

If H Has a Hamiltonian Cycle

Then G has a Hamiltonian Path.

- **Assumption:** H contains a Hamiltonian Cycle, visiting all vertices including Z.
- **Step 1:** Since it's a cycle, it must traverse Z, with edges $u \to Z \to v$.
- **Step 2:** Remove Z and its edges ($u \to Z$ and $Z \to v$).
- **Step 3:** The remaining path $v \to \cdots \to u$ in H consists of edges from G.
- **Step 4:** This path visits all vertices of G (since the cycle in H covered all vertices except Z).
- **Conclusion:** The resulting path in G is a Hamiltonian Path.

Proof of Reduction – Reverse Direction (Visualization)



Remove Z: $v_0 \to v_1 \to \cdots \to v_n$

Proof of Reduction – Conclusion

Summary of the Reduction

The reduction is complete and correct.

- **Forward Direction:** If G has a Hamiltonian Path, H has a Hamiltonian Cycle.
- **Reverse Direction:** If H has a Hamiltonian Cycle, G has a Hamiltonian Path.
- **Implication:** Hamiltonian Path reduces to Hamiltonian Cycle, proving the latter is NP-Hard.
- **NP-Completeness:** Since Hamiltonian Cycle is in NP, it is NP-Complete.

Longest Path Problem

(Maximizing Path Weight Without Vertex Repeats)

Longest Path Problem – Definition

Definition

Given a graph $G = (V, E)$ with edge weights and a number K, determine if there exists a simple path (no repeated vertices) with total weight $\geq K$.

- Applicable to both directed and undirected graphs.
- Focuses on maximizing weight, unlike Hamiltonian Path which requires visiting all vertices.

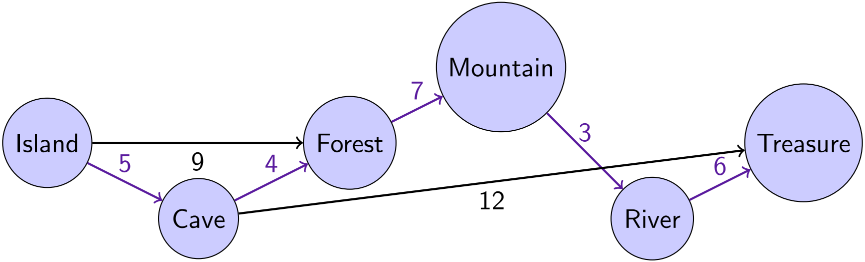
Longest Path Problem – Pirate Adventure

- **Scenario:** A pirate is on a journey to find a hidden treasure. He starts at the Island and can take different routes to the Treasure.
- **Graph:** Vertices: $\{$Island, Cave, Forest, Mountain, River, Treasure$\}$
- **Edges:**

  Island $\rightarrow$ Cave (weight 5), Cave $\rightarrow$ Forest (weight 4), Forest $\rightarrow$ Mountain (w

  River $\rightarrow$ Treasure (weight 6), Island $\rightarrow$ Forest (weight 9), Cave $\rightarrow$ Treasure

- **Goal:** Find the longest path the pirate can take to reach the treasure.



Longest Path Problem – Pirate Adventure (Solution)

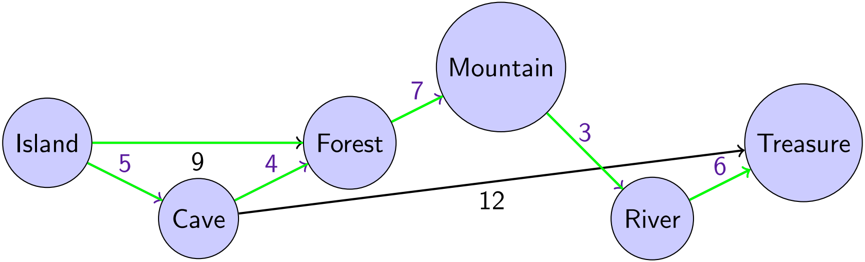
- **Longest Path:** After evaluating all possible paths, the longest path is:

  $$\text{Island} \to \text{Cave} \to \text{Forest} \to \text{Mountain} \to \text{River} \to \text{Treasure}$$

  with a total weight of:

  $$5 + 4 + 7 + 3 + 6 = 25$$

- Another possible path: Island $\to$ Forest $\to$ Mountain $\to$ River $\to$ Treasure, total weight $= 25$.
- The pirate has multiple options, but the total weight for both the paths is 25.



Longest Path Problem – Proving NP-Completeness

Objective

Establish that the Longest Path problem is NP-Complete.

- **In NP:** Given a path (certificate), verify its weight $\geq K$ in polynomial time.
- **NP-Hard:** Demonstrate via a reduction from the Hamiltonian Path problem.

Reduction: Hamiltonian Path to Longest Path

(Using Weights to Verify Hamiltonian Paths)

Reduction: Hamiltonian Path to Longest Path

Goal: Convert an instance of Hamiltonian Path (HP) to an instance of Longest Path (LP)

Given (Input)

- An unweighted graph $G = (V, E)$
- $n = |V|$ (number of vertices)

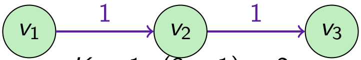
Construct (Output)

- Let $H = G$ (same structure)
- Assign weight $w = 1$ to every edge in H
- Set threshold $K = w \cdot (n - 1) = n - 1$

Why This Works

- A Hamiltonian Path visits all n nodes $\rightarrow$ uses $n - 1$ edges
- All weights are 1 $\rightarrow$ total weight of such a path is $n - 1$
- So: HP exists in G Longest Path in H has weight K

Reduction Setup – Visualization



$$K = 1 \cdot (3 - 1) = 2$$

- Each edge in H has weight 1.
- K is set to match the weight of a path with $n - 1$ edges.

Graph G_1 – Yes Instance

- Nodes: v_1, v_2, v_3
- Edges: $v_1 \rightarrow v_2 \rightarrow v_3$
- Edge weights $= 1$
- $K = 2$, Path weight $= 1 + 1 = 2 \geq K$
- **Yes Instance**

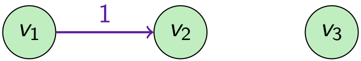
$$v_1 \xrightarrow{1} v_2 \xrightarrow{1} v_3$$

Graph G_2 – Yes Instance

- Nodes: $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$
- Edge weights $= 1$
- $K = 3$, Path weight $= 1 + 1 + 1 = 3 = K$
- **Yes Instance**

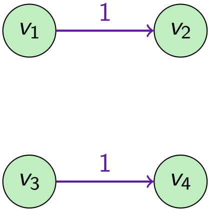
Graph G_3 – No Instance

- Nodes: $v_1 \to v_2$, v_3 is isolated
- Edge weights $= 1$
- $K = 2$, longest path $= 1$ ¡ K
- **No instance**



Graph G_4 – No Instance

- Disconnected components: $v_1 \to v_2$, $v_3 \to v_4$
- Edge weights $= 1$
- $K = 3$, longest path in any component $= 1$ ¡ K
- **No instance**



Proof of Reduction – Forward Direction

Claim: If G has a Hamiltonian Path, then H has a Longest Path of weight $\geq K$

Why?

- A Hamiltonian Path in G visits all n vertices exactly once
- This path must use $n-1$ edges
- In H, all edges have weight $w = 1$
- So the path's total weight is:

$$\text{Weight} = (n-1) \cdot 1 = n - 1$$

- But $K = n - 1$ by construction
- Therefore, the path in H has weight $\geq K$

Conclusion: Hamiltonian Path $\Rightarrow$ Longest Path (weight $\geq K$)

Proof of Reduction – Reverse Direction (Contraposition)

If H Has a Longest Path with Weight $\geq K$

Then G has a Hamiltonian Path.

- **Assumption:** Assume H has a path with weight $\geq K = n - 1$.
- **Weight and Edges:** Since all edges in H have weight 1, the path must have at least $n - 1$ edges.
- **Forming the Hamiltonian Path in G:** A path with $n - 1$ edges necessarily visits n distinct vertices, making it a Hamiltonian Path in G.

Conclusion: A Longest Path in H with weight $\geq n - 1$ corresponds to a Hamiltonian Path in G.

$$v_1 \xrightarrow{1} v_2 \xrightarrow{1} v_3$$

Weight $= 2 \geq K$, $n - 1$ edges

Proof of Reduction – Conclusion

What We've Shown

We reduced the Hamiltonian Path problem to the Longest Path problem. This proves that Longest Path is **NP-Hard**.

- **Forward Direction:**
  - If G has a Hamiltonian Path
  - Then H has a path of weight $\geq K$
- **Reverse Direction:**
  - If H has a path with weight $\geq K$
  - Then that path must visit n nodes, i.e., a Hamiltonian Path in G
- **Conclusion:**
  - Longest Path is NP-Hard
  - Since it's also in NP $\rightarrow$ **Longest Path is NP-Complete**

Final Result: Longest Path is NP-Complete

Traveling Salesperson Problem (TSP)

(Finding a Tour with Minimum Weight)

TSP – Definition

Definition

Given a weighted graph $G = (V, E, w)$ and a bound m, find a tour that starts at any vertex, visits all vertices at least once, returns to the start, with total weight $\leq m$.

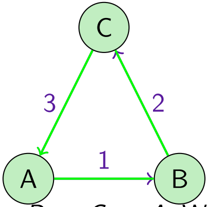
- Allows revisiting vertices or edges unless specified otherwise.

TSP – Tour Structure and Example

Key Rules

Start at any vertex, visit all vertices, return to the starting vertex. The total weight is the sum of edge weights in the tour.

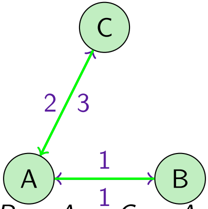
- **Example:** Vertices $\{A, B, C\}$; Edges: $A \to B$ (1), $B \to C$ (2), $C \to A$ (3).
- **Tour:** $A \to B \to C \to A$, weight $= 1 + 2 + 3 = 6$.



Tour: $A \to B \to C \to A$, Weight $= 6$

TSP – Revisiting Vertices

- **Revisiting Example:** Same graph with vertices $\{A, B, C\}$.
- **Tour with Revisits:** $A \to B \to A \to C \to A$.
- **Weight:** $A \to B\ (1) + B \to A\ (1) + A \to C\ (2) + C \to A\ (3) = 7$.
- **Observation:** Revisiting increases the tour's weight.



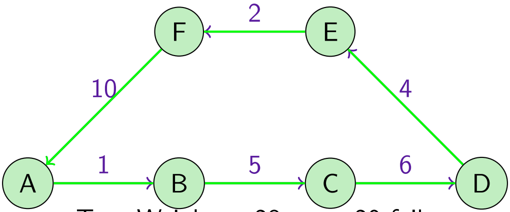
Tour: $A \to B \to A \to C \to A$, Weight $= 7$

TSP – Larger Example

Complex Graph

Graph with 6 vertices $\{A, B, C, D, E, F\}$.

- **Edges:** $A \to B$ (1), $B \to C$ (5), $C \to D$ (6), $D \to E$ (4), $E \to F$ (2), $F \to A$ (10).
- **Tour:** $A \to B \to C \to D \to E \to F \to A$.
- **Weight:** $1 + 5 + 6 + 4 + 2 + 10 = 28$.
- **Check:** Fails if $m = 20$.



Tour Weight $= 28$, $m = 20$ fails

Reduction: Hamiltonian Cycle to TSP

(Proving NP-Hardness of TSP)

Reduction: Hamiltonian Cycle to TSP (Setup)

Goal

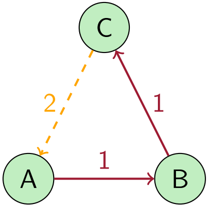
Convert a Hamiltonian Cycle problem into a Traveling Salesman Problem (TSP) instance.

Step-by-Step Transformation:

- **Start with a graph** $G = (V, E)$ where you want to check if a Hamiltonian Cycle exists.
- **Create a new graph** H that is a complete graph with the same set of vertices V.
- **Assign edge weights in** H**:**
  - If an edge exists in G, set weight $= 1$
  - If an edge does not exist in G, set weight $= 2$
- **Set the TSP cost bound** $m = n$, where $n =$ number of vertices.

This ensures: TSP tour of weight n *exists if and only if* G *has a Hamiltonian Cycle.*

Reduction Setup – Visualization



- Original edges from G have weight 1.
- Added edges to complete the graph have weight 2.

Reduction Logic – Connecting Hamiltonian Cycle to TSP

Core Idea

A TSP tour in H with a cost $\leq n$ can only use the edges from the original graph G.

- To complete a tour, we need at least n edges (one for each vertex) and then return to the start.
- If the tour's total cost is $\leq n$, each edge must have weight 1, because any edge with weight 2 would make the cost exceed n.
- Therefore, a tour with total cost $\leq n$ must only use the edges that exist in G, which means the tour forms a Hamiltonian Cycle in G.

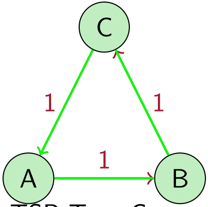
Conclusion: A valid TSP tour with cost n directly corresponds to a Hamiltonian Cycle in G.

Proof – Forward Direction

If G Has a Hamiltonian Cycle

Then H has a TSP tour with cost $\leq n$.

- A Hamiltonian Cycle in G uses n edges, all with weight 1 in H.
- This cycle becomes a TSP tour in H with cost n.
- Since $m = n$, the tour satisfies the TSP condition.



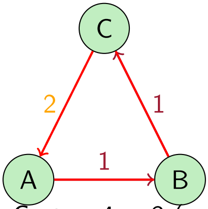
H.C. in G, TSP Tour Cost $= 3$ ($n = 3$)

Proof – Reverse Direction

If G Has No Hamiltonian Cycle

Then H has no TSP tour with cost $\leq n$.

- Without a Hamiltonian Cycle in G, any tour in H must use at least one edge with weight 2.
- A tour with at least one weight-2 edge has cost $\geq (n-1) \cdot 1 + 2 = n + 1 > n$.
- Thus, no tour in H can have cost $\leq n$.



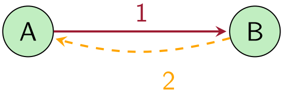
Tour Cost $= 4 > 3$ $(n = 3)$

TSP Example – Graph G_1

Graph G_1

Vertices: $\{A, B\}$; Edge: $A \to B$ (weight 1 in H).

- **H.P.:** $A \to B$ (yes).
- **H.C.:** None (no return edge).
- H_1**:** Complete graph, $A \to B$ (1), $B \to A$ (2).
- **TSP Tour:** $A \to B \to A$, cost $= 1 + 2 = 3 > 2$ ($n = 2$).



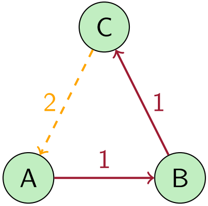
H.P. Yes, H.C. No, TSP Cost $= 3$

TSP Example – Graph G_2

Graph G_2

Vertices: $\{A, B, C\}$; Edges: $A \to B$ (1), $B \to C$ (1).

- **H.P.:** $A \to B \to C$ (yes).
- **H.C.:** None (no $C \to A$ edge).
- H_2**:** Complete graph with $C \to A$ added (2).
- **TSP Tour:** $A \to B \to C \to A$, cost $= 1 + 1 + 2 = 4 > 3$ $(n = 3)$.



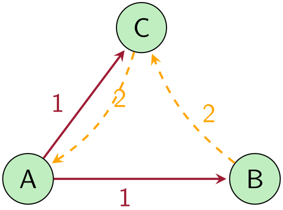
H.P. Yes, H.C. No, TSP Cost $= 4$

TSP Example – Graph G_3

Graph G_3

Vertices: $\{A, B, C\}$; Edges: $A \to B$ (1), $A \to C$ (1).

- **H.P.:** None (cannot visit all vertices in one path).
- **H.C.:** None (no cycle exists).
- H_3**:** Complete graph with $B \to C$ (2), $C \to A$ (2).
- **TSP Tour:** $A \to B \to C \to A$, cost $= 1 + 2 + 2 = 5 > 3$ ($n = 3$).



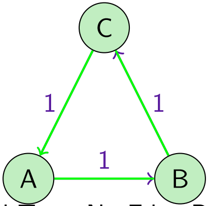
H.P. No, H.C. No, TSP Cost = 5

TSP Variant – No Edge Reuse

Variant 1: No Edge Reuse

A TSP variant where edges cannot be used more than once in the tour.

- The tour must be a simple cycle, akin to a Hamiltonian Cycle.
- **NP-Completeness:** Directly reducible from the Hamiltonian Cycle problem.



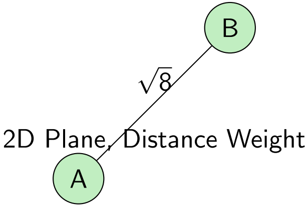
Valid Tour, No Edge Reuse

TSP Variant – Euclidean Distances

Variant 2: Euclidean Distances

Vertices are points in a 2D plane, with edge weights as Euclidean distances.

- **Example:** Points $A(0, 0)$, $B(2, 2)$, distance $= \sqrt{8}$.
- **NP-Completeness:** Reducible from the Hamiltonian Cycle problem.



NP-Completeness of TSP

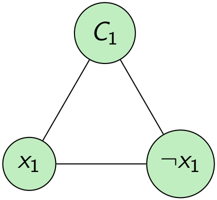
Why NP-Complete?

TSP is both in NP and NP-Hard, making it NP-Complete.

- **In NP:** Verify a tour's weight $\leq m$ in $O(n)$ time.
- **NP-Hard:** Proven via reduction from Hamiltonian Cycle.
- **Variants:** Both "No Edge Reuse" and "Euclidean Distances" variants are also NP-Complete.

Exercise – 3-SAT to 2-Colorability

- **Task:** Reduce 3-SAT to 2-Colorability.
- **2-Colorability:** Color a graph with 2 colors such that no adjacent vertices share the same color.
- **Objective:** Construct a graph where a 2-coloring exists if the 3-SAT formula is satisfiable.
- **Example Graph:**



Exercise – 3-SAT to 4-Colorability

- **Task:** Summarize the reduction from 3-SAT to 4-Colorability.
- **Formula:** $(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3)$.
- **Objective:** Explain how OR gadgets ensure a 4-coloring reflects a satisfying assignment.
- **Details:** OR gadgets create subgraphs where colorings correspond to true literals in the 3-SAT formula.

Exercise – Hamiltonian Path to TSP

- **Task:** Reduce Hamiltonian Path to TSP.
- **Approach:**
  - Construct a complete graph H from G.
  - Set weights: $w(e) = 1$ if $e \in G$, otherwise $w(e) = 2$.
  - Set bound $m = n - 1$.
- **Challenge:** Ensure the TSP tour corresponds to a Hamiltonian Path in G.

Exercise – Longest Path Reduction Verification

- **Task:** Verify the Hamiltonian Path to Longest Path reduction with a new graph.
- **Graph:** Choose a graph with 5 vertices, some with a Hamiltonian Path and some without.
- **Objective:** Apply the reduction, compute K, and check the Longest Path outcome.

Key Takeaways

- **NP-Completeness:** Problems like Hamiltonian Cycle, Longest Path, and TSP are in NP and NP-Hard.
- **Hamiltonian Path and Cycle:** Reduction involves adding vertex Z, with a complete proof using contraposition.
- **Longest Path:** Reduced from Hamiltonian Path using uniform weights and $K = n - 1$.
- **TSP:** Reduced from Hamiltonian Cycle with weights 1 and 2, $m = n$, variants are NP-Complete.
- **Verification:** Polynomial-time verification using certificates (e.g., vertex lists, divisors).

Key Concepts – NP-Completeness Diagram

- **NP-Completeness Recap:**
  - **In NP:** Verifiable in polynomial time.
  - **NP-Hard:** At least as hard as the hardest NP problems.
  - **NP-Complete:** Both in NP and NP-Hard.

Summary

- **Hamiltonian Path:** Successfully identified $F \to D \to B \to A \to C \to E$.
- **Hamiltonian Cycle:** Proved NP-Complete via reductions from 3-SAT and Hamiltonian Path.
- **Longest Path:** Established NP-Completeness through reduction from Hamiltonian Path.
- **TSP:** Reduced from Hamiltonian Cycle, explored variants, confirmed NP-Completeness.
- **Verification:** Demonstrated polynomial-time verification for NP problems.

Next Steps

- Explore detailed reductions, e.g., 3-SAT to Hamiltonian Cycle.
- Practice additional reductions, such as Hamiltonian Path to Longest Path on new graphs.
- Investigate optimization techniques for TSP.
- Implement algorithms for Hamiltonian Path and TSP in Python.
- Questions or topics for further exploration?