國立臺灣大學
National Taiwan University

CHAPTER 8
BEYOND POLYNOMIAL RUNNING TIMES

Iris Hui-Ru Jiang
Fall 2017

Department of Electrical Engineering
National Taiwan University

Outline

- Content:
  - Polynomial-time reduction
  - NP-completeness
- Reading:
  - Chapter 8

The Status of the P Versus NP Problem
http://cacm.acm.org/magazines/2009/9/38904-the-status-of-the-p-versus-np-problem/fulltext

Easy vs. Hard

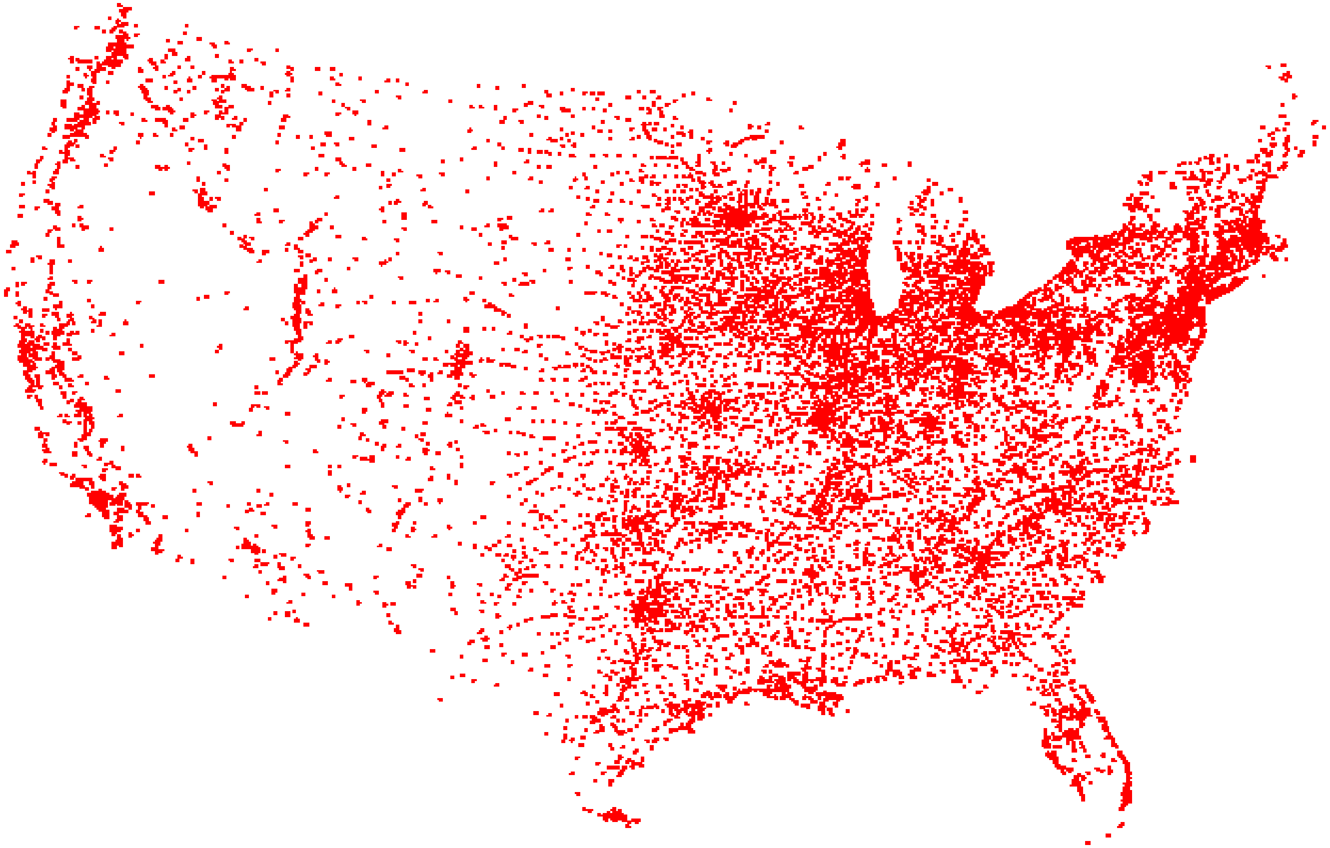
- Q: Which problems will we be able to solve in practice?
- A working definition.
  - [Cobham 1964, Edmonds 1965, Rabin 1966] Those with polynomial-time algorithms.

- Desiderata: Classify problems according to those that can be solved in polynomial-time and those that cannot.
- Provably requires exponential-time.
- Frustrating news: Huge number of fundamental problems have defined classification for decades.
- Chapter 8: Show that these fundamental problems are computationally equivalent and appear to be different manifestations of one really hard problem.

Decision & Optimization Problems

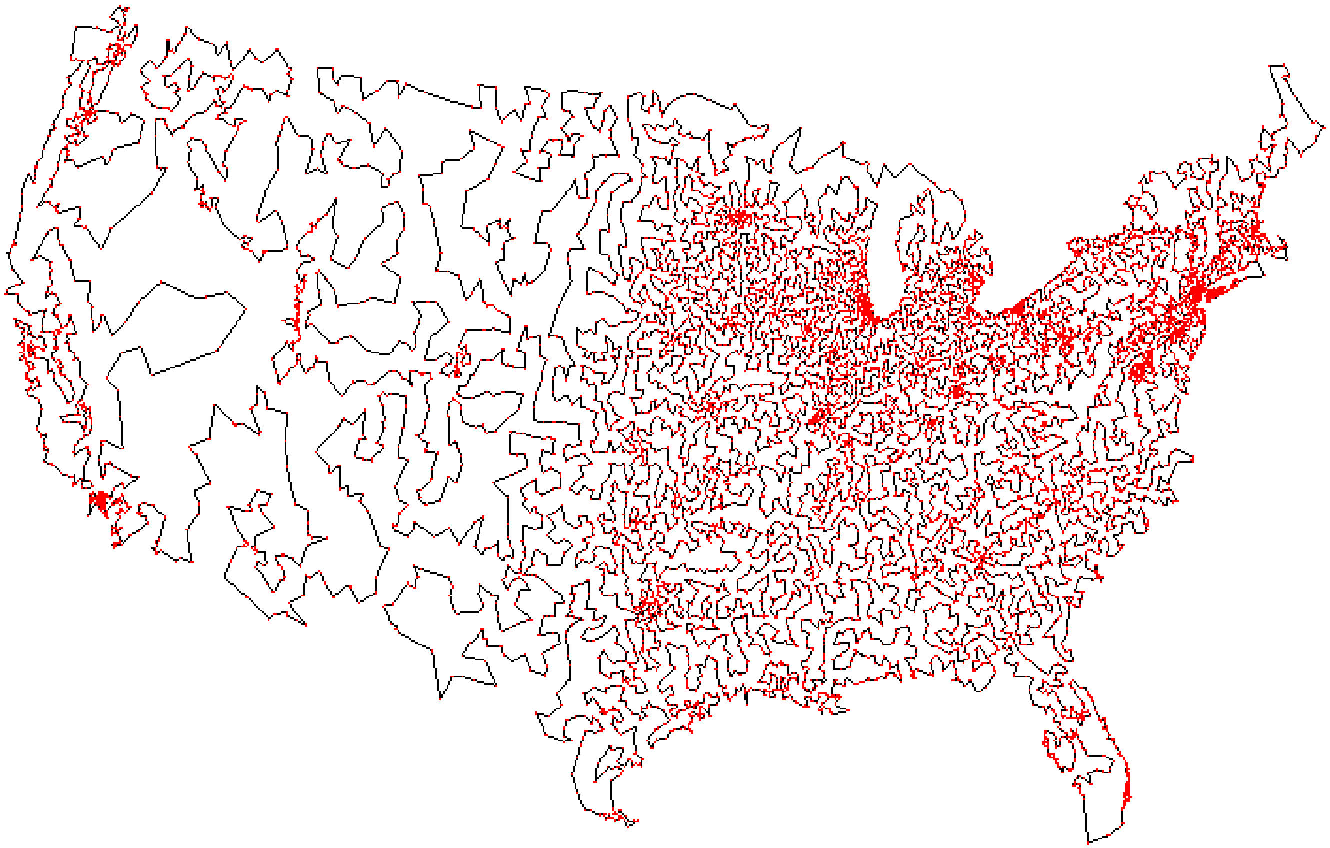
- Decision problems: those having yes/no answers.
  – MST: Given a graph $G=(V, E)$ and a bound K, is there a spanning tree with a cost at most K?
  – TSP: Given a set of cities, distance between each pair of cities, and a bound B, is there a route that starts and ends at a given city, visits every city exactly once, and has total distance at most B?
- Optimization problems: those finding a legal configuration such that its cost is minimum (or maximum).
  – MST: Given a graph $G=(V, E)$, find the cost of a minimum spanning tree of G.
  – TSP: Given a set of cities and that distance between each pair of cities, find the distance of a "minimum route" starts and ends at a given city and visits every city exactly once.
- Could apply binary search on a decision problem to obtain solutions to its optimization problem.
- Class NP is associated with decision problems.

Traveling Salesman Problem (TSP) (1/2)



All 13,509 cities in US with a population of at least 500

Traveling Salesman Problem (TSP) (2/2)



Optimal TSP tour

Complexity Classes

- Developed by S. Cook and R. Karp in early 1970.
- The class P: class of problems that can be solved in polynomial time in the size of input.
- The class NP (Nondeterministic Polynomial): class of problems that can be verified in polynomial time in the size of input.
  - P=NP?
- The class NP-complete (NPC): A problem Y in NP with the property that for every problem X in NP, $X \leq_p Y$.
- Theorem: Suppose Y is NPC, then Y is solvable in polynomial time iff P = NP.
  - Any NPC problem can be solved in polynomial time $\Rightarrow$ All problems in NP can be solved in polynomial time.
- Fundamental question: Do there exist “natural” NPC problems?

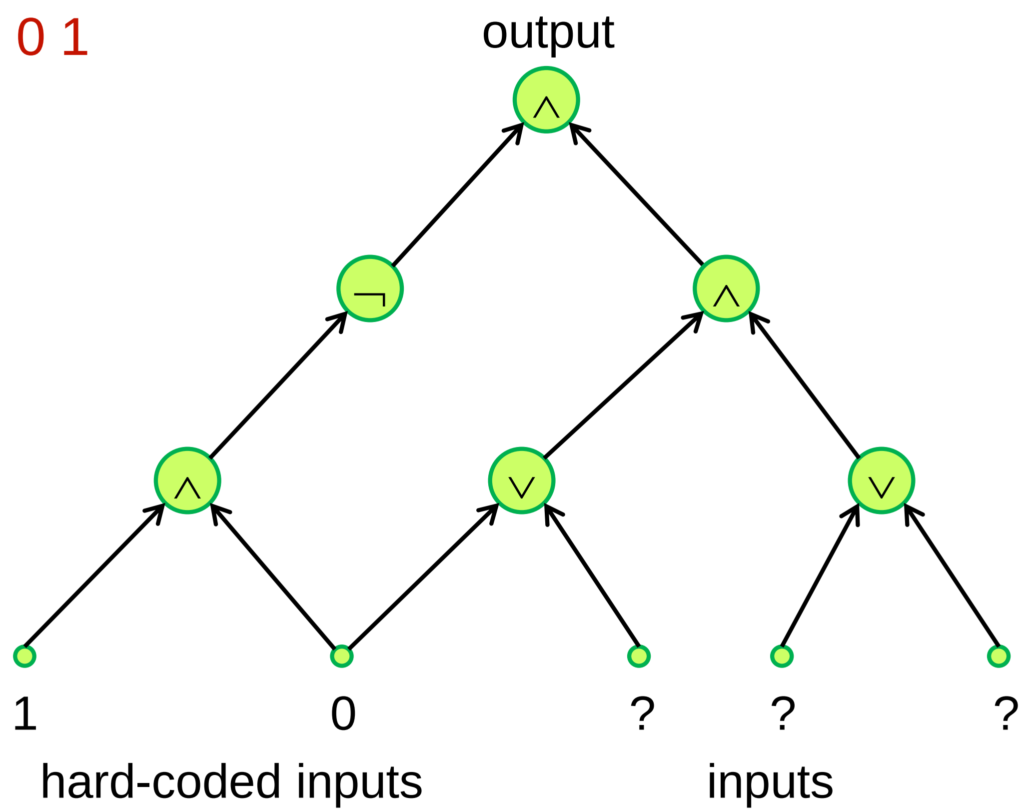
Verification Algorithm and Class NP

- Verification algorithm: a 2-argument algorithm *A*, where one argument is an input string *x* and the other is a binary string *y* (called a certificate). *A* verifies *x* if there exists *y* s.t. *A* answers "yes."

- Ex: The Traveling Salesman Problem (TSP)
  - **Instance:** a set of *n* cities, distance between each pair of cities, and a bound *B*.
  - **Question:** is there a route that starts and ends at a given city, visits every city exactly once, and has total distance $\leq B$?
- Is TSP $\in$ NP?
- Need to check a solution in polynomial time.
  - Guess a tour (certificate).
  - Check if the tour visits every city exactly once.
  - Check if the tour returns to the start.
  - Check if total distance $\leq B$.
- All can be done in $O(n)$ time, so TSP $\in$ NP.

The First Proved NPC: Circuit Satisfiability

- CIRCUIT-SAT:
  - Q: Given a combinational circuit built out of AND, OR, and NOT gates, is there a way to set the circuit inputs so that the output is 1?
  - A: Yes: 1 0 1



More Hard Computational Problems

- Aerospace engineering: optimal mesh partitioning for finite elements.
- Biology: protein folding.
- Chemical engineering: heat exchanger network synthesis.
- Civil engineering: equilibrium of urban traffic flow.
- Economics: computation of arbitrage in financial markets with friction.
- Electrical engineering: VLSI layout.
- Environmental engineering: optimal placement of contaminant sensors.
- Financial engineering: find minimum risk portfolio of given return.
- Game theory: find Nash equilibrium that maximizes social welfare.
- Genomics: phylogeny reconstruction.
- Mechanical engineering: structure of turbulence in sheared flows.
- Medicine: reconstructing 3-D shape from biplane angiocardiogram.
- Operations research: optimal resource allocation.
- Physics: partition function of 3-D Ising model in statistical mechanics.
- Politics: Shapley-Shubik voting power.
- Pop culture: Minesweeper consistency.
- Statistics: optimal experimental design.

Polynomial-Time Reduction (1/2)

- Desiderata: Suppose we could solve Y in polynomial-time. What else could we solve in polynomial time?

- Reduction: Problem X polynomial reduces to problem Y if given an arbitrary instance x of problem X, we can construct an input y to problem Y in polynomial time such that x is a yes instance to X iff y is a yes instance of Y.
  - Notation: $X \leq_P Y$.

- Remarks:
  - The algorithm for Y is viewed as a black box.
  - We pay for polynomial time to write down instances sent to this black box

Polynomial-Time Reduction (2/2)

- Purpose: Classify problems according to relative difficulty.

1. Design algorithms: If X $\leq_P$ Y and Y can be solved in polynomial-time, then X can also be solved in polynomial time.
   - Bipartite matching $\leq_P$ Network flow
2. Establish intractability: If X $\leq_P$ Y and X cannot be solved in polynomial-time, then Y cannot be solved in polynomial time.
   - Hamiltonian cycle $\leq_P$ Travelling salesman
3. Establish equivalence: If X $\leq_P$ Y and Y $\leq_P$ X, X $\equiv_P$ Y.
   - Up to cost of reduction

Coping with a "Tough" Problem: Trilogy I

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

Courtesy of Prof. Y.-W. Chang

Coping with a “Tough” Problem: Trilogy II

“I can’t find an efficient algorithm, because no such algorithm is possible!”

Courtesy of Prof. Y.-W. Chang

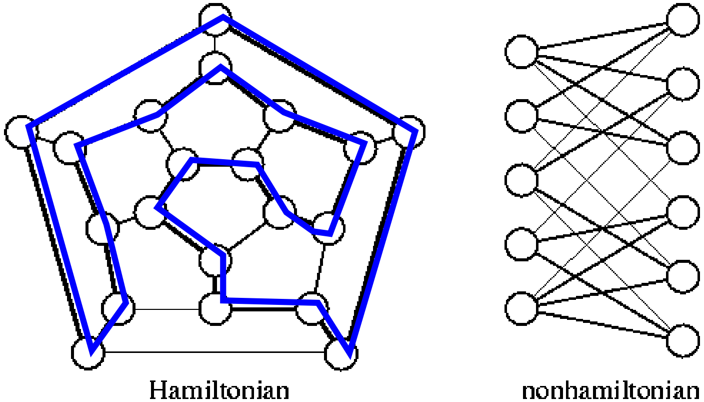
Coping with a "Tough" Problem: Trilogy III

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

Courtesy of Prof. Y.-W. Chang

Polynomial Reduction: HC $\leq_P$ TSP

- The Hamiltonian Circuit Problem (HC)
  - **Instance:** an undirected graph $G = (V, E)$.
  - **Question:** is there a cycle in G that includes every vertex exactly once?
- TSP: The Traveling Salesman Problem
- Claim: HC $\leq_P$ TSP.
  1. Define a function f mapping **any** HC instance into a TSP instance, and show that f can be computed in polynomial time.
  2. Prove that G has an HC iff the reduced instance has a TSP tour **with distance $\leq B$** ($x \in$ HC $\Leftrightarrow f(x) \in$ TSP).

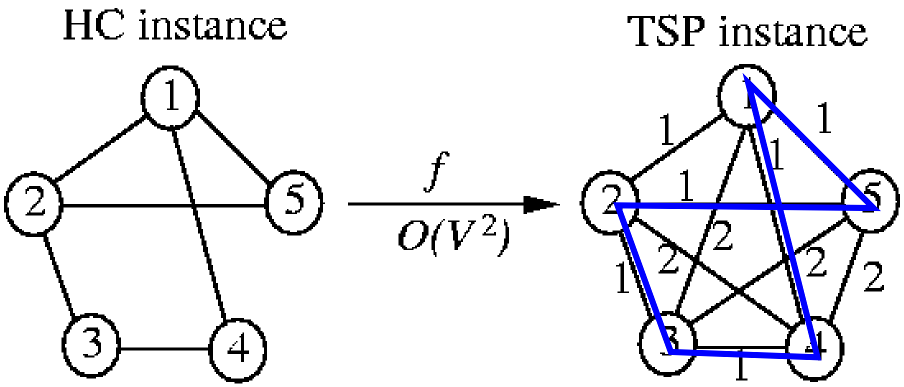


HC $\leq_P$ TSP: Step 1

- Define a reduction function *f* for HC $\leq_P$ TSP.
  - Given an HC instance $G = (V, E)$ with n vertices
    - Create a set of n cities labeled with names in V.
    - Assign distance between u and v

$$d(u,v) = \begin{cases} 1, & \text{if } (u,v) \in E, \\ 2, & \text{if } (u,v) \notin E. \end{cases}$$

    - Set bound $B = n$.
  - f can be computed in $O(V^2)$ time.

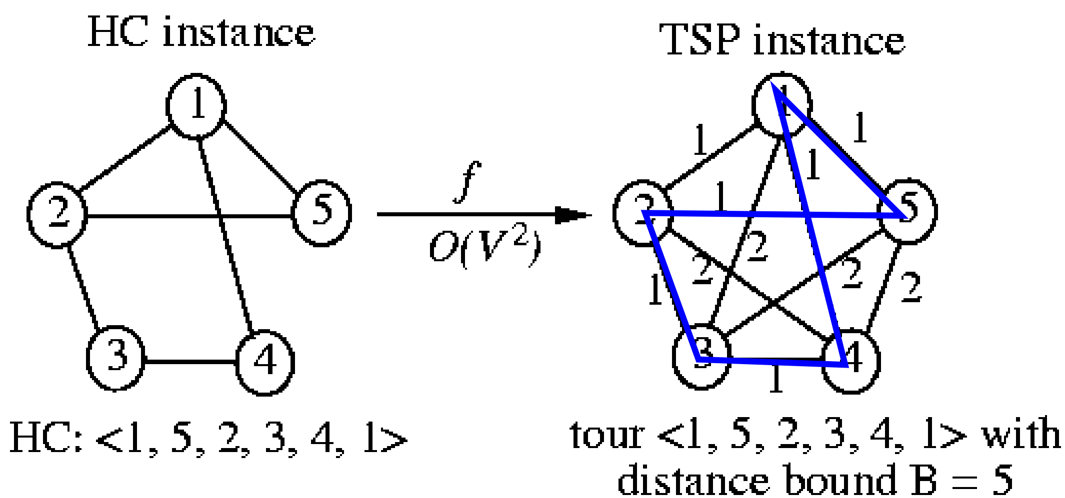


HC: <1, 5, 2, 3, 4, 1>

tour <1, 5, 2, 3, 4, 1> with distance bound B = 5

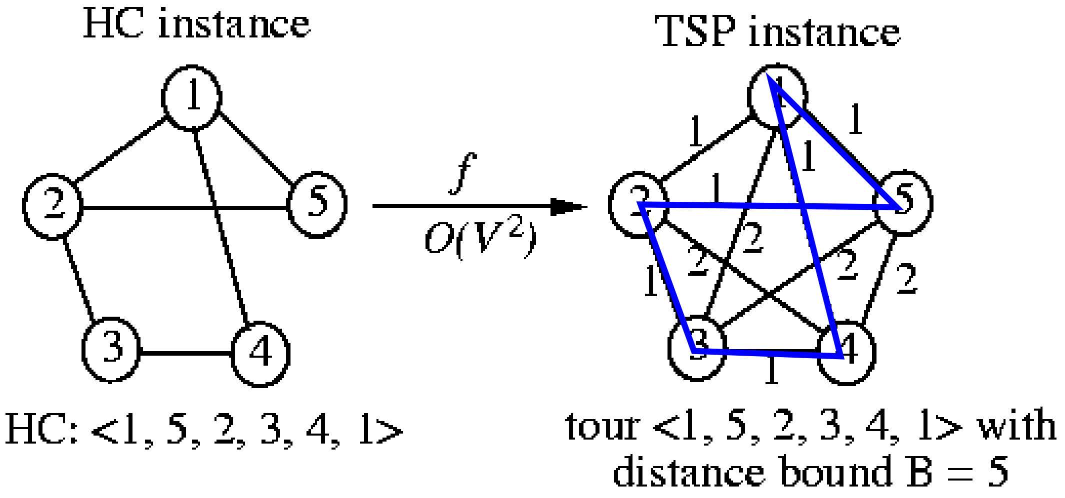
HC $\leq_P$ TSP: Step 2

- *G* has a HC iff the reduced instance has a TSP with distance $\leq B$.
  - $x \in$ HC $\Rightarrow f(x) \in$ TSP.
    - Suppose the HC is $h = \langle v_1, v_2, \ldots, v_n, v_1 \rangle$. Then, *h* is also a tour in the transformed TSP instance.
    - The distance of the tour *h* is $n = B$ since there are *n* consecutive edges in *E*, and so has distance 1 in $f(x)$.
    - Thus, $f(x) \in$ TSP ($f(x)$ has a TSP tour with distance $\leq B$).



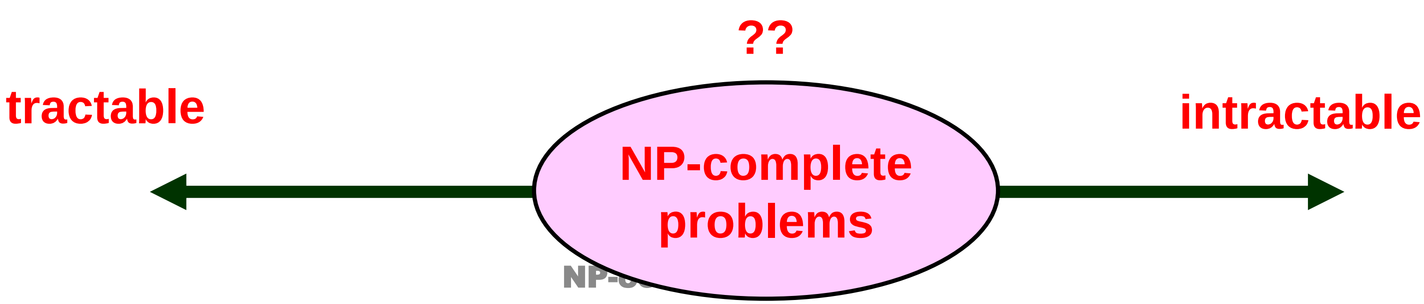
HC $\leq_P$ TSP: Step 2 (cont'd)

- *G* has a HC iff the reduced instance has a TSP with distance $\leq B$.
  - $f(x) \in$ TSP $\Rightarrow x \in$ HC.
    - Suppose there is a TSP tour with distance $\leq n = B$. Let it be $<v_1, v_2, \ldots, v_n, v_1>$.
    - Since distance of the tour $\leq n$ and there are n edges in the TSP tour, the tour contains only edges in E.
    - Thus, $<v_1, v_2, \ldots, v_n, v_1>$ is a Hamiltonian cycle ($x \in$ HC).



NP-Completeness

- Definition: A decision problem L (a language $L \subseteq \{0, 1\}^*$) is NP-complete (NPC) if
  1. $L \in$ NP, and
  2. $L' \leq_P L$ for every $L' \in$ NP.
- NP-hard: If L satisfies property 2, but not necessarily property 1, we say that L is NP-hard.
- Suppose $L \in$ NPC. P=NP?
  - If $L \in P$, then there exists a polynomial-time algorithm for every $L' \in$ NP (i.e., P = NP).
  - If $L \notin P$, then there exists no polynomial-time algorithm for any $L' \in$ NPC (i.e., P $\neq$ NP).



Proving NP-Completeness

- Theorem: A decision problem L (a language $L \subseteq \{0, 1\}^*$) is NP-complete (NPC) if
  1. $L \in$ NP, and
  2. $L' \leq_P L$ for an $L' \in$ NPC.
- Five steps for proving that L is NP-complete:
  1. Prove $L \in$ NP.
  2. Select a known NP-complete problem L'.
  3. Construct a reduction f transforming **every** instance of L' to an instance of L.
  4. Prove that f is a polynomial-time transformation.
  5. Prove that $x \in L'$ iff $f(x) \in L$ for all $x \in \{0, 1\}^*$.