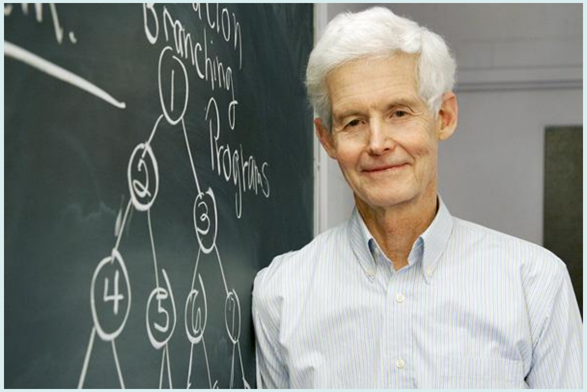




CSE 417
Algorithms and Complexity

Autumn 2024

Lecture 28

NP-Completeness

Announcements

- Homework 9
- Exam practice problems on course homepage
- Final Exam: Monday, December 9, 8:30 AM
  - One Hour Fifty Minutes
  - Closed book

~~Mon, Dec 2~~	~~NP-Completeness~~
Wed, Dec 4	NP-Completeness
Fri, Dec 6	Last Lecture: NP-Completeness and Beyond
Mon, Dec 9	Final Exam

Exam Format

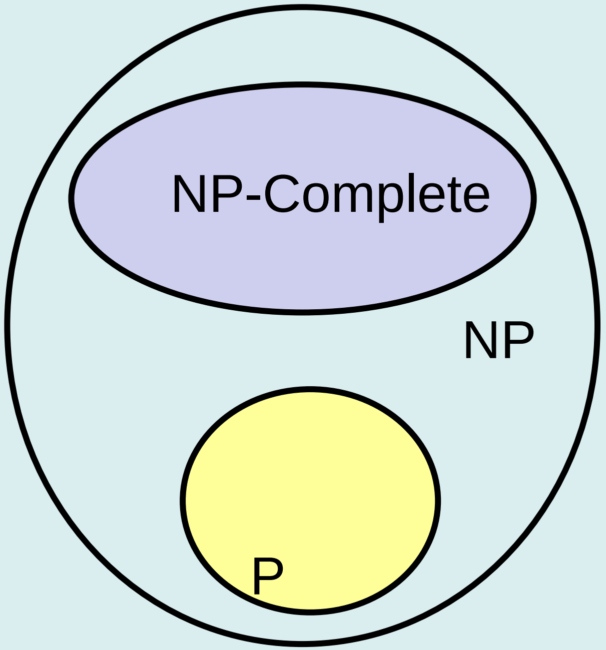
- Two short answer problems
- Six additional problems

- Potential Exam Questions
  - Recurrences
  - Divide and Conquer
  - One dimensional dynamic programming
  - Two dimensional dynamic programming
  - Network Flow / MaxFlow-MinCut
  - Reductions to Network Flow

- Material: 25% pre-midterm, 75% post-midterm

The Universe

- P: Polynomial Time
- NP: Nondeterministic Polynomial Time
  - Problems where a "yes" answer can be verified in polynomial time
- NP-Complete
  - The hardest problems in NP



Polynomial time reductions

- X is Polynomial Time Reducible to Y
  - Solve problem X with a polynomial number of computation steps and a polynomial number of calls to a black box that solves Y
  - Notations: $X <_P Y$

- Usually, this is converting an input of X to an input for Y, solving Y, and then converting the answer back

Lemmas

- If $X <_P Y$ and $Y <_P Z$ then $X <_P Z$

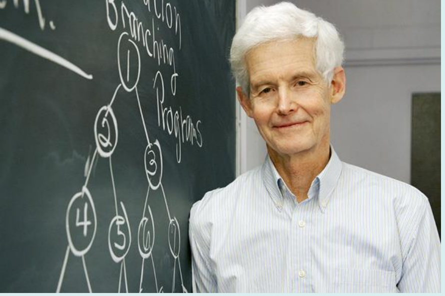
- Suppose $X <_P Y$. If Y can be solved in polynomial time, then X can be solved in polynomial time.

- Suppose $X <_P Y$. If X cannot be solved in polynomial time, then Y cannot be solved in polynomial time.

NP-Completeness

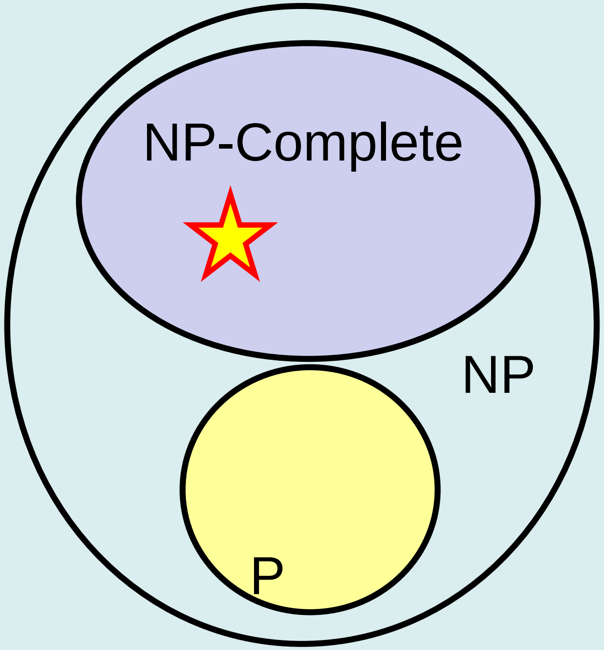
- A problem X is NP-complete if
  - X is in NP
  - For every Y in NP, $Y <_P X$
- X is a "hardest" problem in NP

If X is NP-Complete, Z is in NP and $X <_P Z$ then Z is NP-Complete



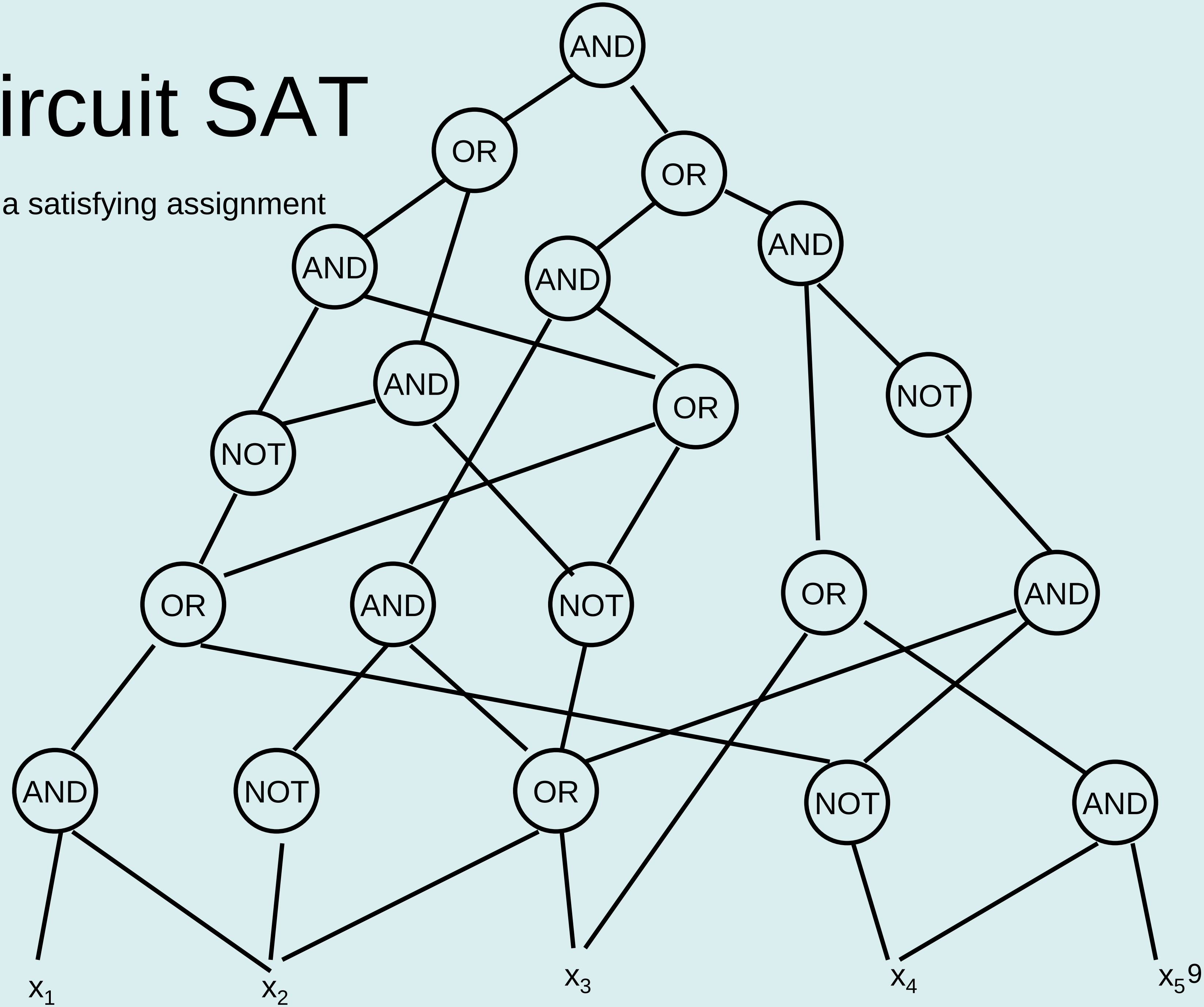
Cook’s Theorem

- There is an NP Complete problem
  - The Circuit Satisfiability Problem

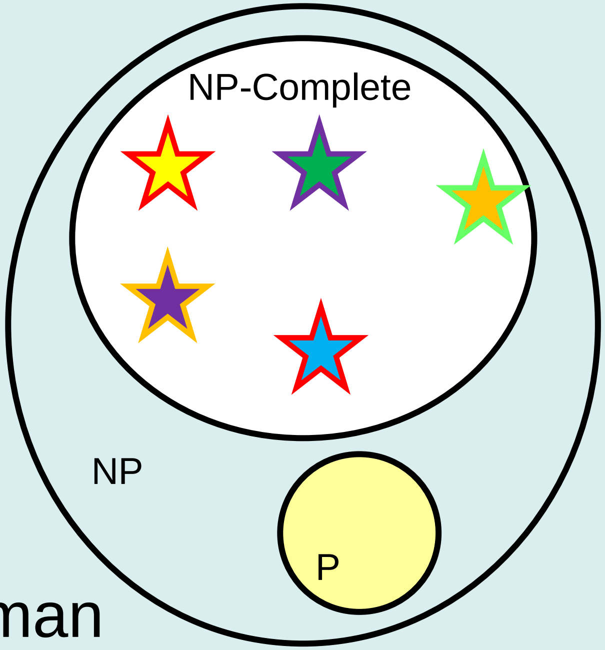


Circuit SAT

Find a satisfying assignment



Populating the NP-Completeness Universe



- Circuit Sat $<_P$ 3-SAT
- 3-SAT $<_P$ Independent Set
- 3-SAT $<_P$ Vertex Cover
- Independent Set $<_P$ Clique
- 3-SAT $<_P$ Hamiltonian Circuit
- Hamiltonian Circuit $<_P$ Traveling Salesman
- 3-SAT $<_P$ Integer Linear Programming
- 3-SAT $<_P$ Graph Coloring
- 3-SAT $<_P$ Subset Sum
- Subset Sum $<_P$ Scheduling with Release times and deadlines

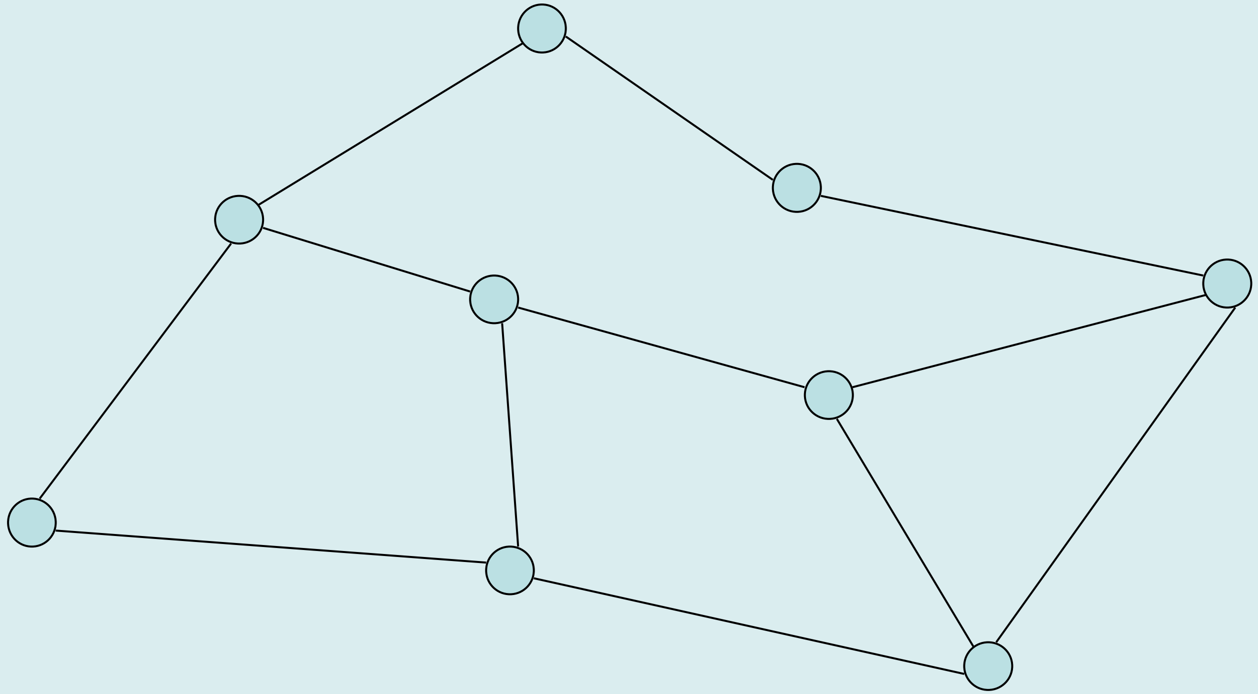
NP Completeness Proofs

If X is NP-Complete, Z is in NP and X $<_P$ Z then Z is NP-Complete

- Pick a known NP-Complete problem and develop a reduction
- Two common types of reductions
  - Modification based (generally easy)
  - Gadget based from SAT (generally not easy)
- Make sure you have the direction of the reduction correct
  - Known NPC problem $<_P$ your problem

Graph Coloring

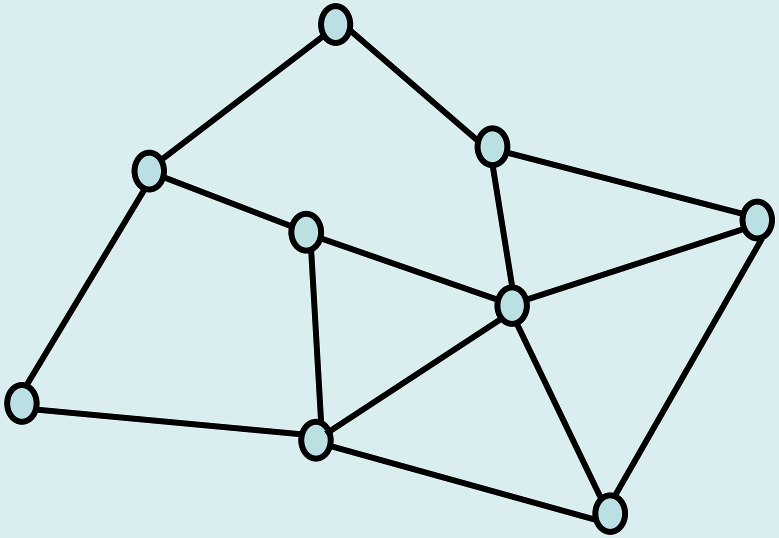
- NP-Complete
  - Graph 3-coloring
- Polynomial
  - Graph 2-Coloring

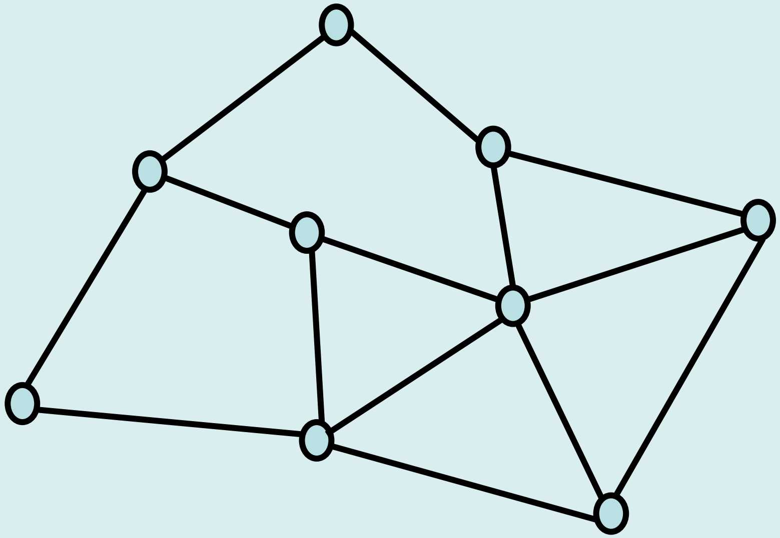


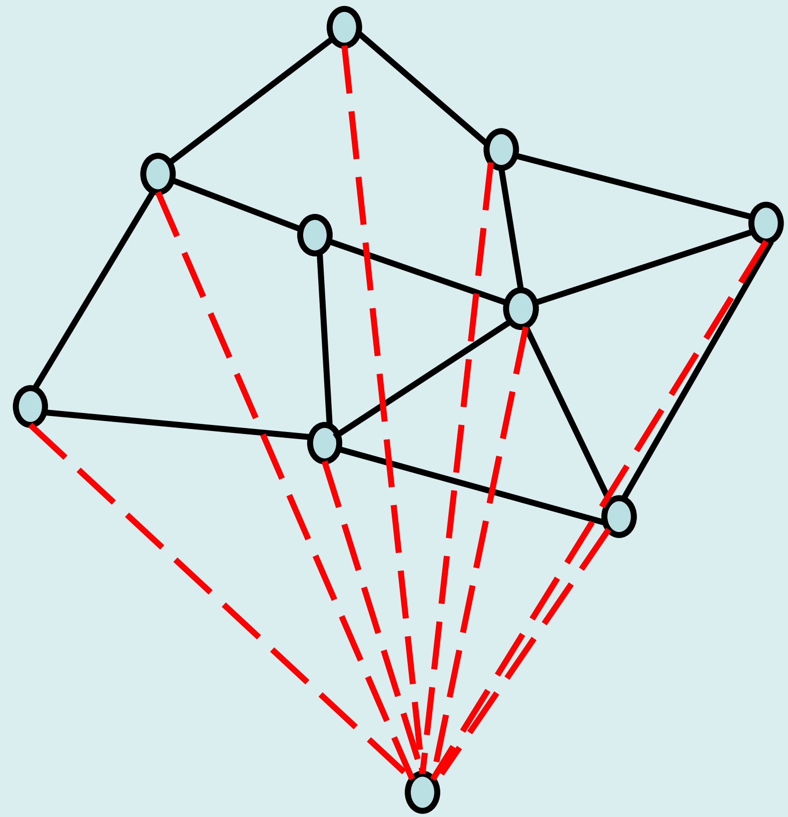
Graph 4-Coloring

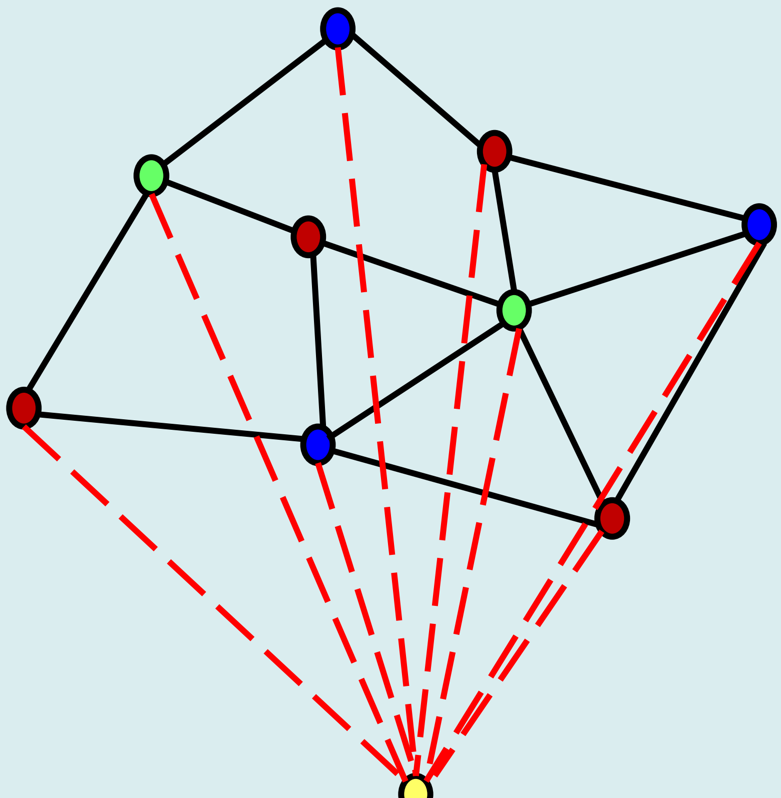
- Given a graph G, can G be colored with 4 colors?
- Prove 4-Coloring is NP Complete
- Proof: 3-Coloring $<_P$ 4-Coloring
- Show that you can 3-Color a graph if you have an algorithm to 4-Color a graph

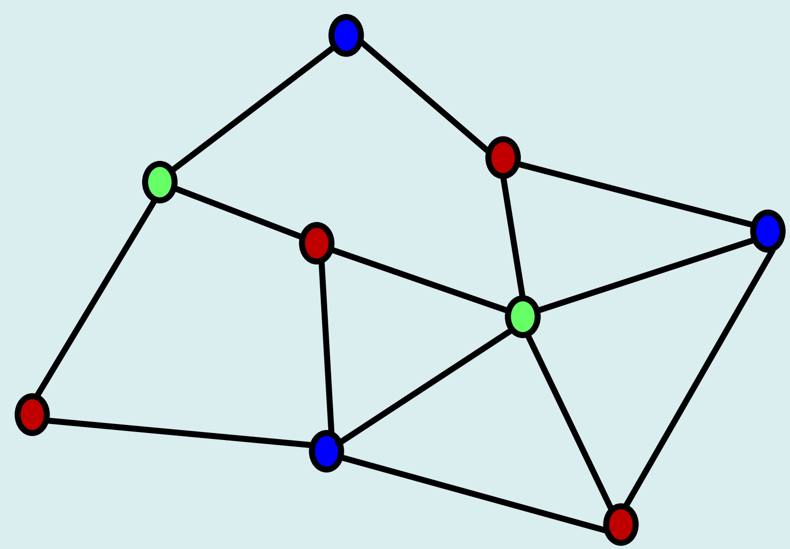
3-Coloring $<_P$ 4-Coloring



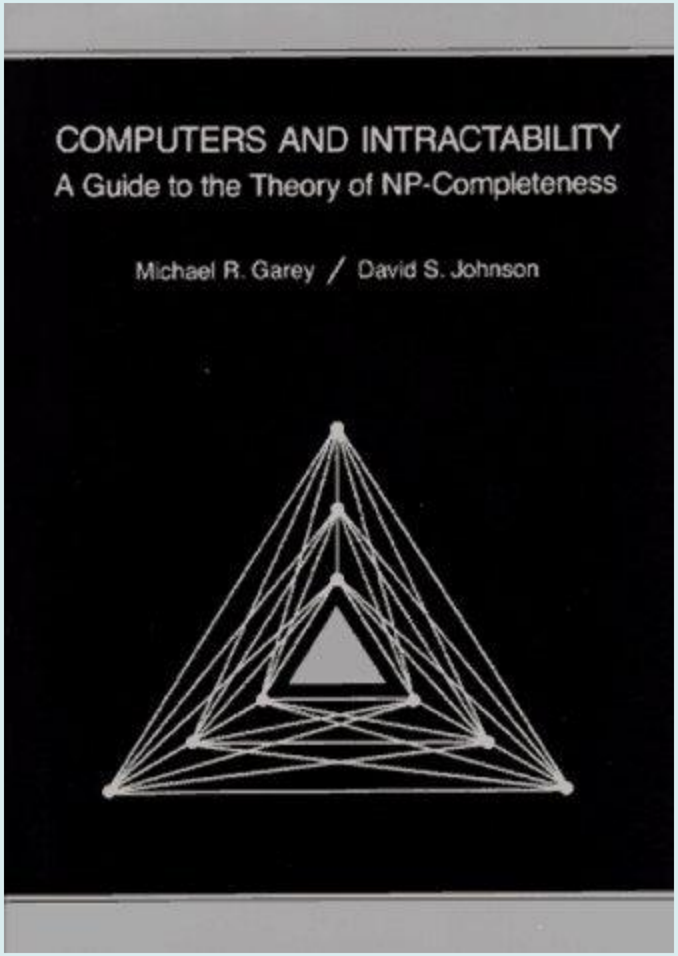






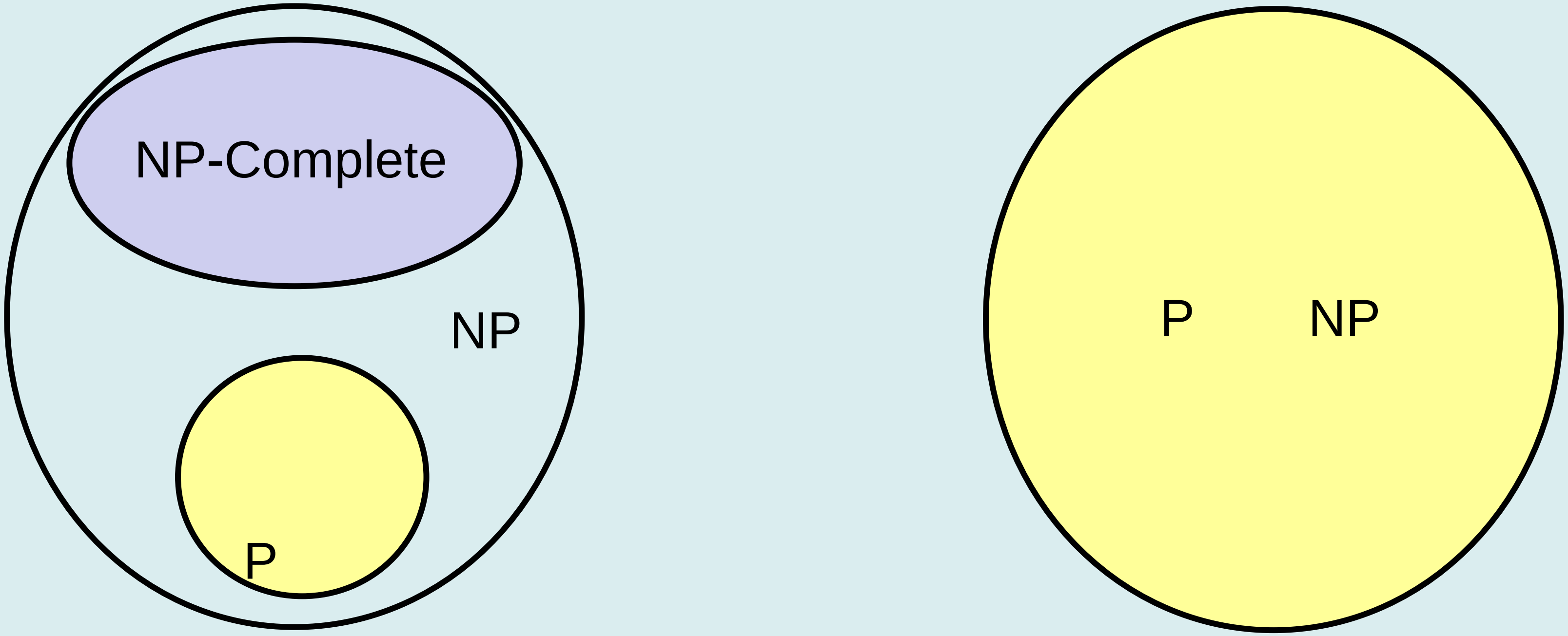


Garey and Johnson



List price: $159.99
Amazon: $109.05
Kindle: $9.99
eBay Used: $10.00 - $30.00

P vs. NP Question



How to prove P = NP

If X is NP-Complete and X can be solved in polynomial time, then P = NP

Satisfiability

Literal: A Boolean variable or its negation. $x_i \text{ or } \overline{x_i}$

Clause: A disjunction of literals. $C_j = x_1 \vee \overline{x_2} \vee x_3$

Conjunctive normal form: A propositional formula Φ that is the conjunction of clauses. $\Phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

SAT: Given CNF formula Φ, does it have a satisfying truth assignment?

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

Ex: $\left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(x_2 \vee x_3\right) \wedge \left(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}\right)$

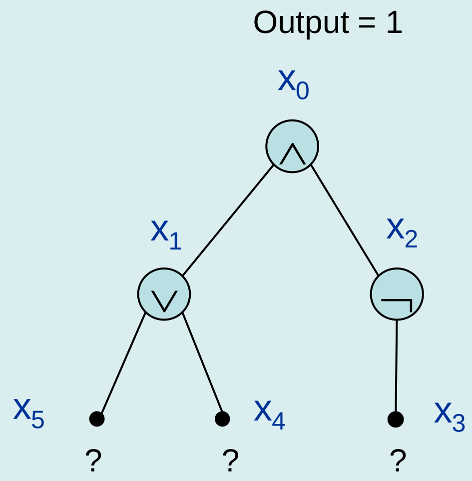
Yes: x_1 = true, x_2 = true x_3 = false.

3-SAT is NP-Complete

Circuit SAT $<_P$ 3-SAT

Convert a circuit into a formula

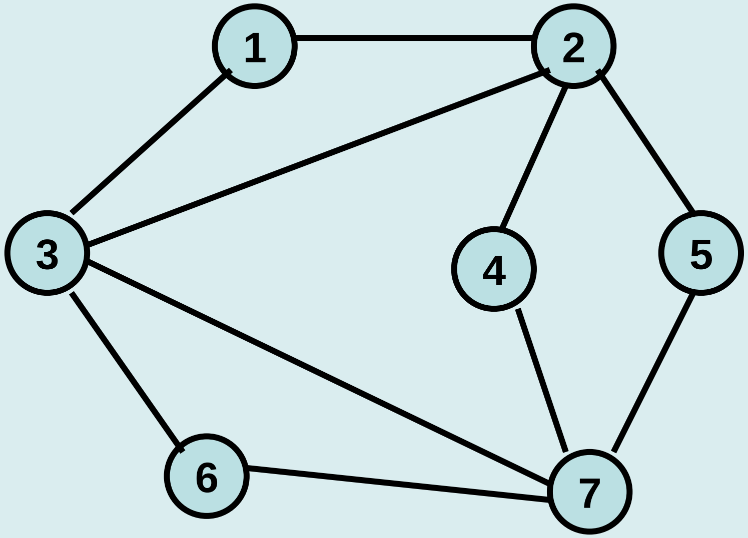
Each gate is represented by a set of clauses



$$(x_1 \vee x_1 \vee x_1) \wedge (x_2 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_1 \vee x_1 \vee \overline{x_4}) \wedge (x_1 \vee x_1 \vee \overline{x_5}) \wedge (\overline{x_1} \vee x_4 \vee x_5) \wedge (\overline{x_0} \vee \overline{x_0} \vee x_1) \wedge (\overline{x_0} \vee \overline{x_0} \vee x_2) \wedge (x_0 \vee \overline{x_1} \vee \overline{x_2})$$

Independent Set

- Independent Set
  - Graph G = (V, E), a subset S of the vertices is independent if there are no edges between vertices in S



3 Satisfiability Reduces to Independent Set

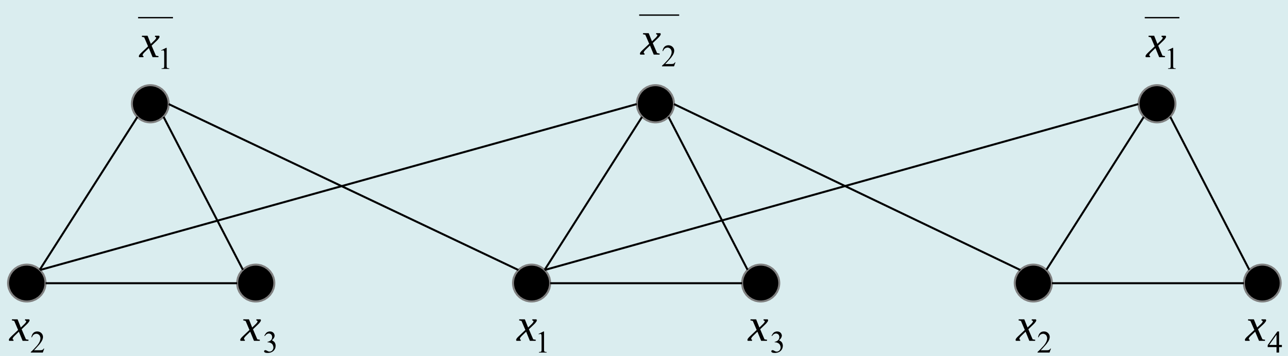
Claim. 3-SAT $\leq_P$ INDEPENDENT-SET.

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

Construction.

- G contains 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.

G

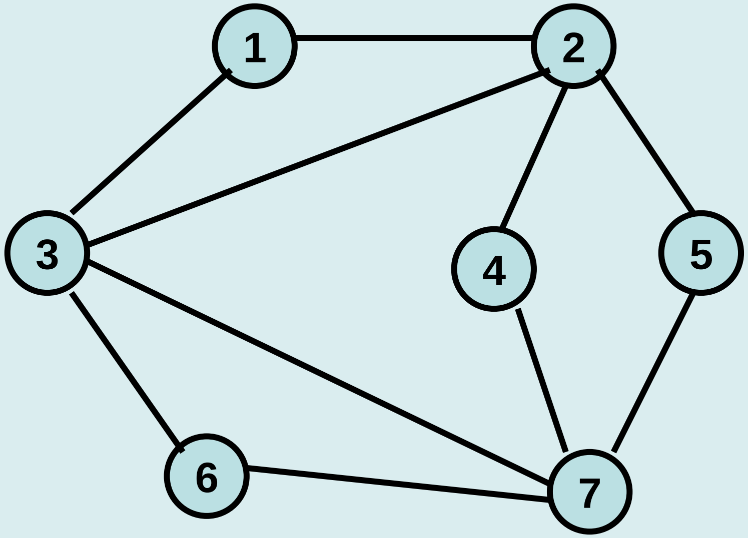


k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

Vertex Cover

- Vertex Cover
  - Graph G = (V, E), a subset S of the vertices is a vertex cover if every edge in E has at least one endpoint in S



Problem: Does G have a vertex cover of size at most K?

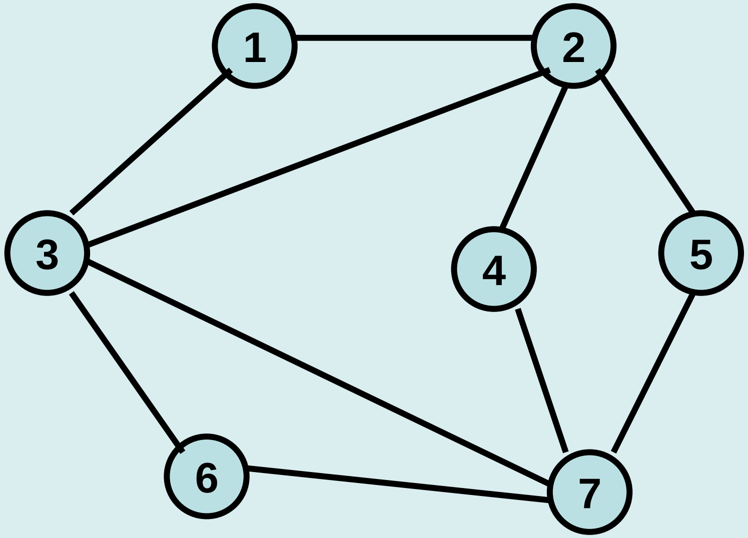
IS $<_P$ VC

- Lemma: A set S is independent iff V-S is a vertex cover

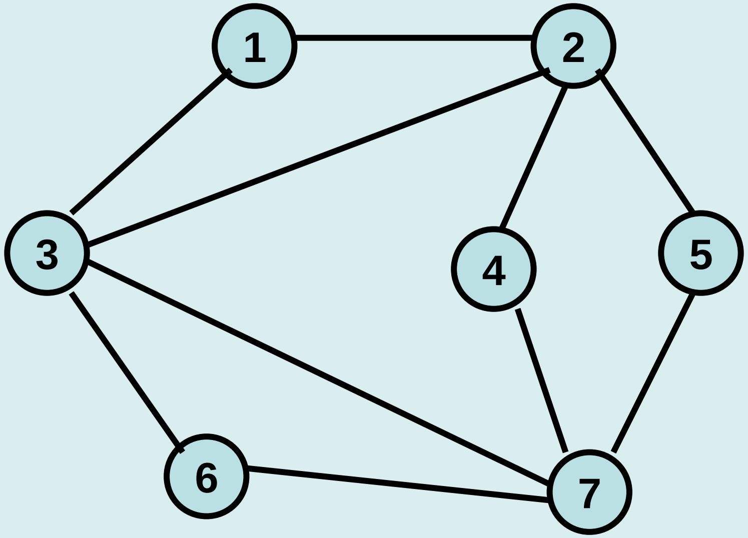
- To reduce IS to VC, we show that we can determine if a graph has an independent set of size K by testing for a Vertex cover of size n - K

IS $<_P$ VC

Find a maximum independent set S

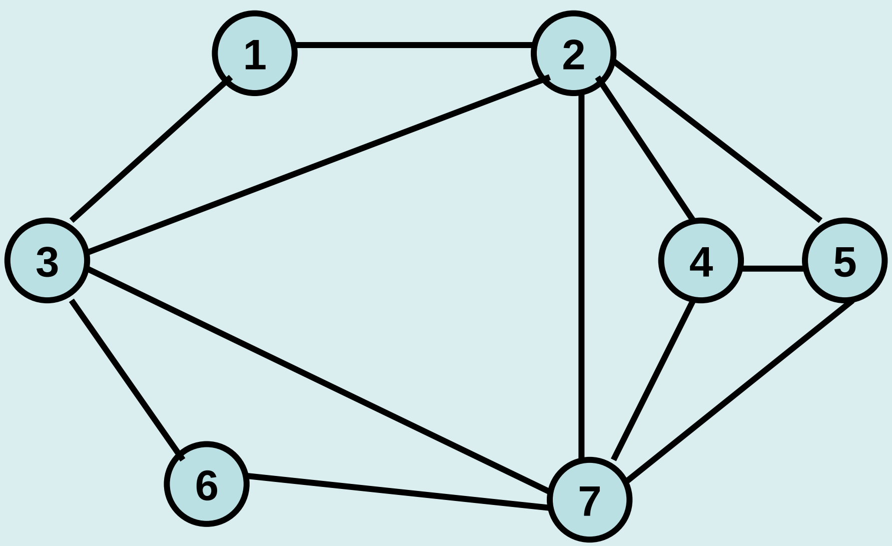


Show that V-S is a vertex cover



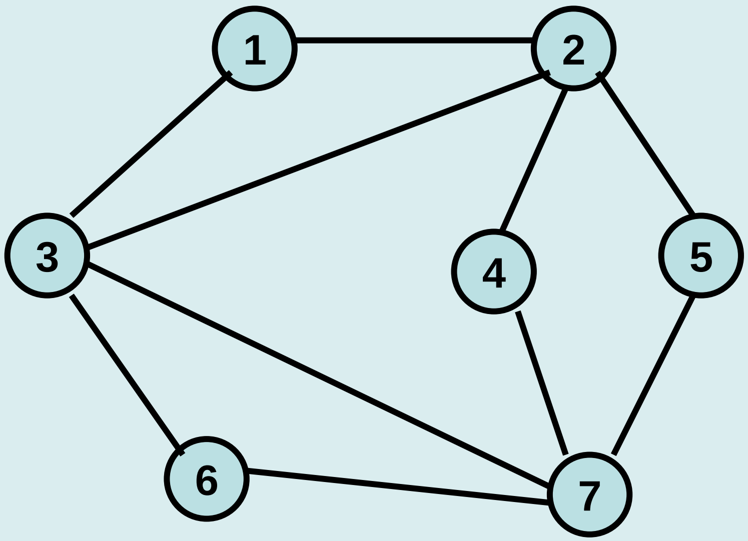
Clique

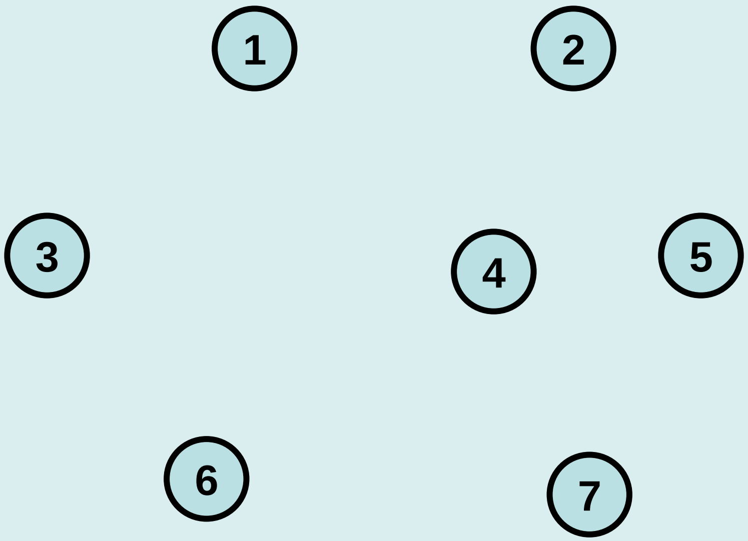
- Clique
  - Graph G = (V, E), a subset S of the vertices is a clique if there is an edge between every pair of vertices in S



Complement of a Graph

- Defn: G’=(V,E’) is the complement of G=(V,E) if (u,v) is in E’ iff (u,v) is not in E





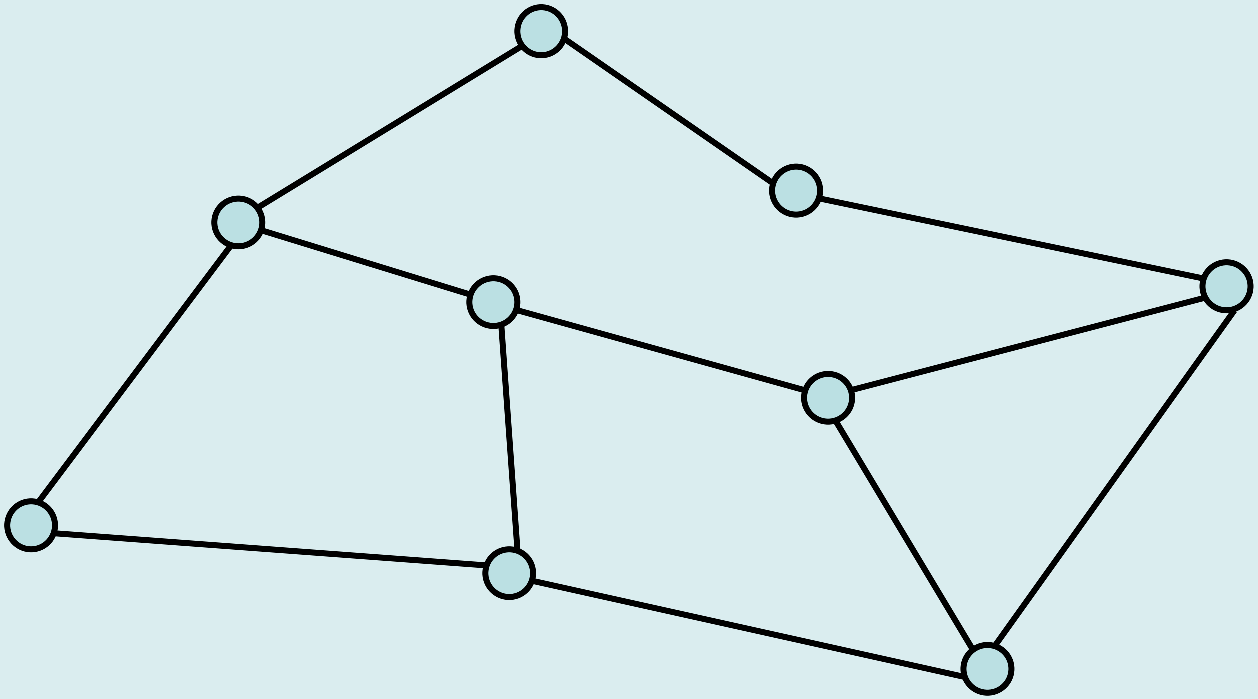
IS $<_P$ Clique

- Lemma: S is Independent in G iff S is a Clique in the complement of G

- To reduce IS to Clique, we compute the complement of the graph. The complement has a clique of size K iff the original graph has an independent set of size K

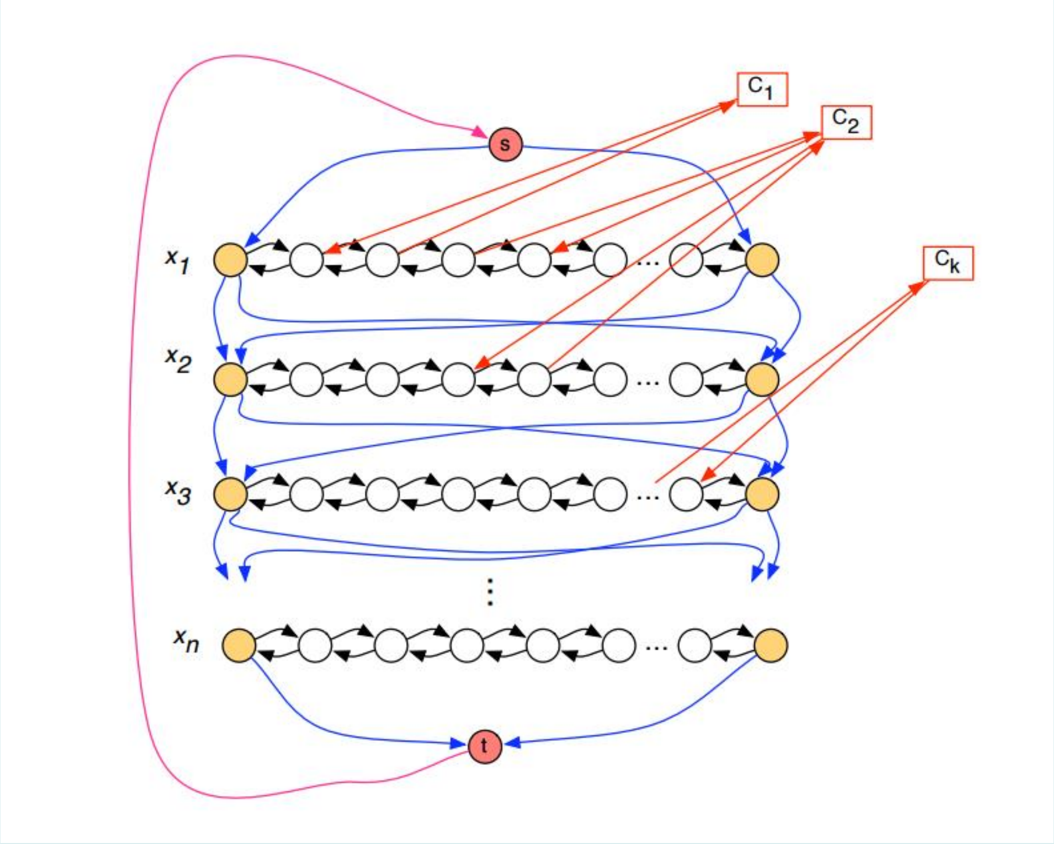
Hamiltonian Circuit Problem

- Hamiltonian Circuit – a simple cycle including all the vertices of the graph



Thm: Hamiltonian Circuit is NP Complete

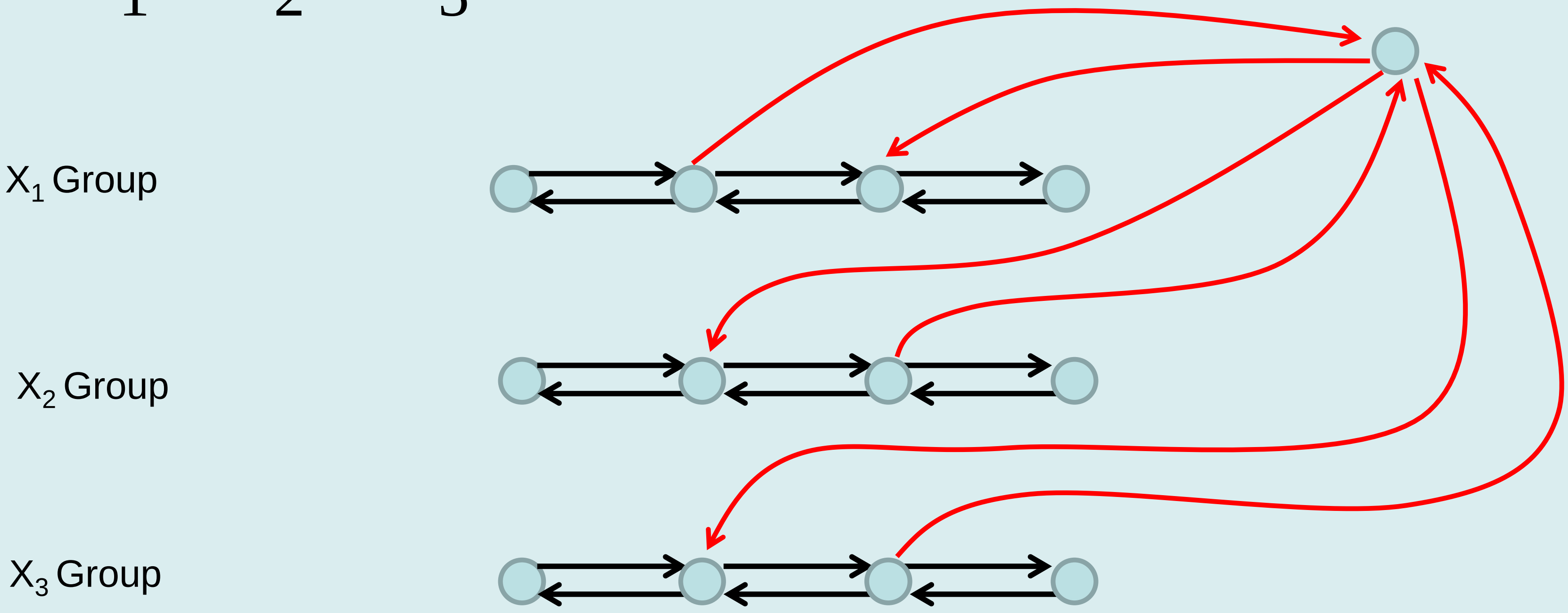
- Reduction from 3-SAT



Text: page 475

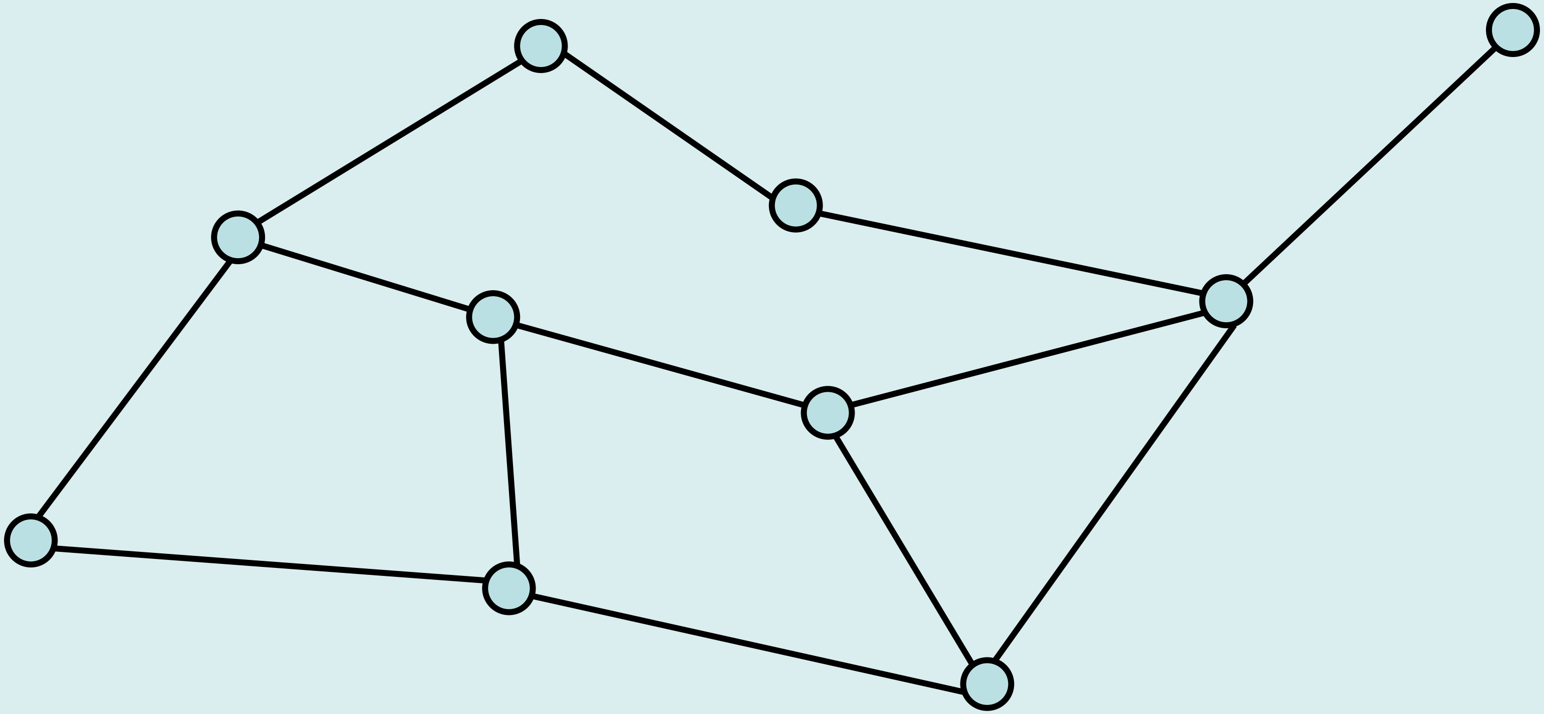
Clause Gadget

$$x_1 \vee \overline{x_2} \vee \overline{x_3}$$



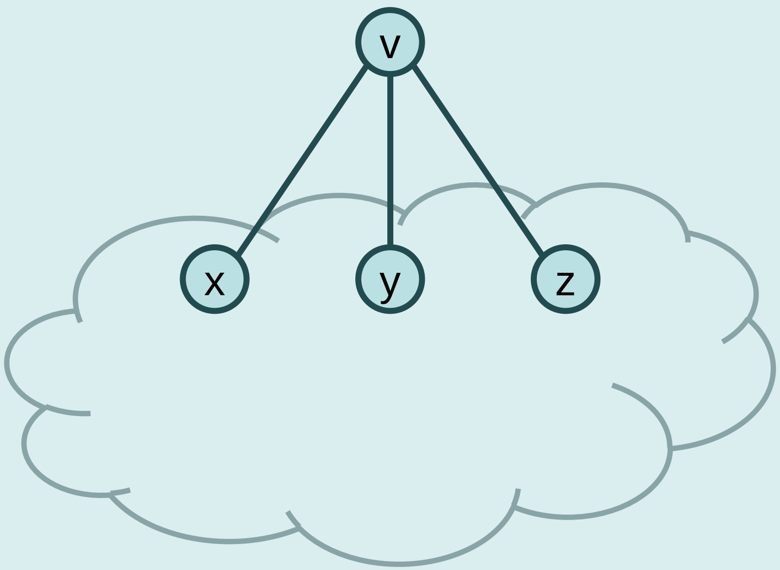
Hamiltonian Path Problem

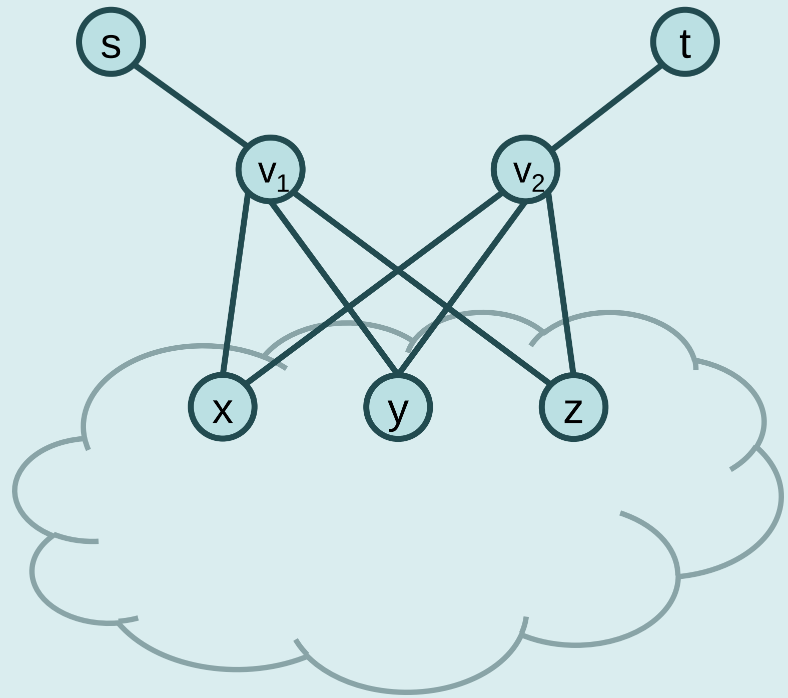
- Hamiltonian Path – a simple path including all the vertices of the graph



Reduce Hamiltonian Circuit to Hamiltonian Path

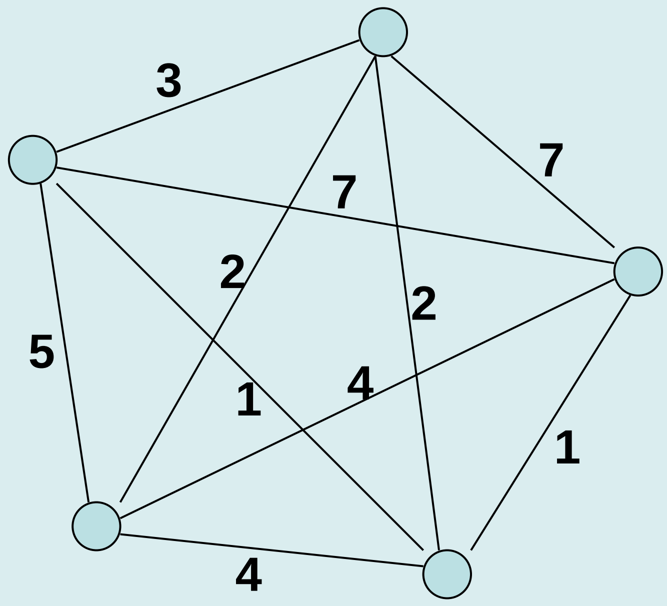
G_2 has a Hamiltonian Path iff G_1 has a Hamiltonian Circuit





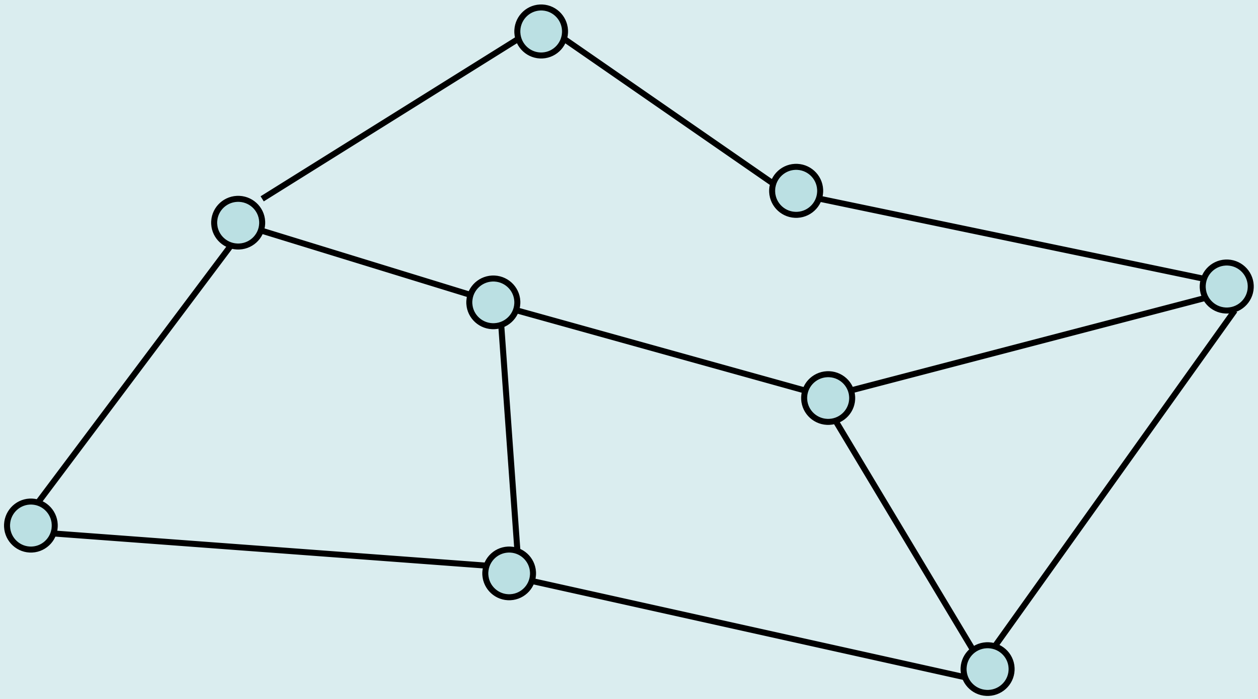
Traveling Salesman Problem

- Given a complete graph with edge weights, determine the shortest tour that includes all of the vertices (visit each vertex exactly once, and get back to the starting point)

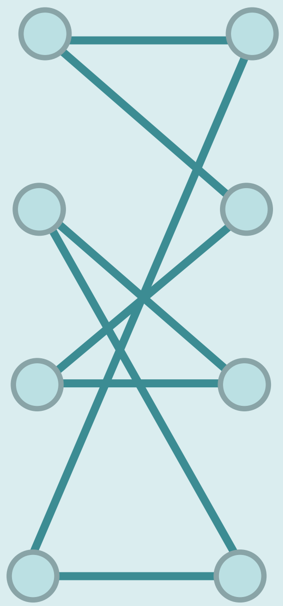


Find the minimum cost tour

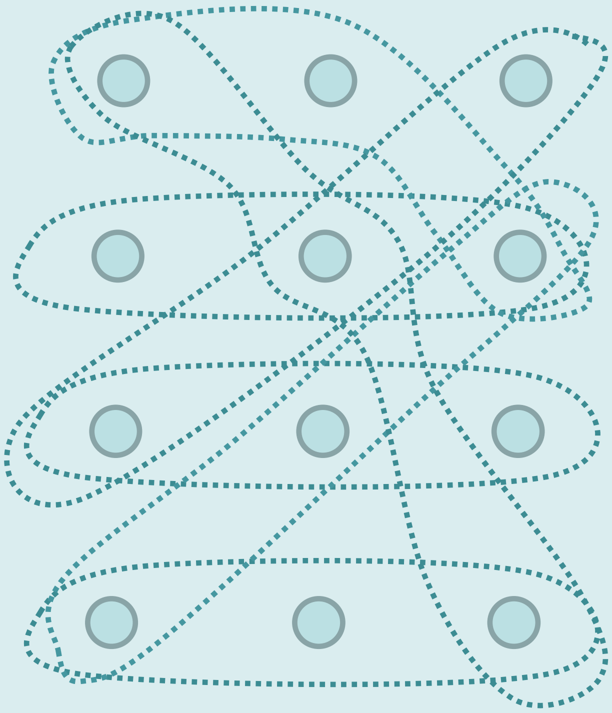
Thm: HC $<_P$ TSP



Matching

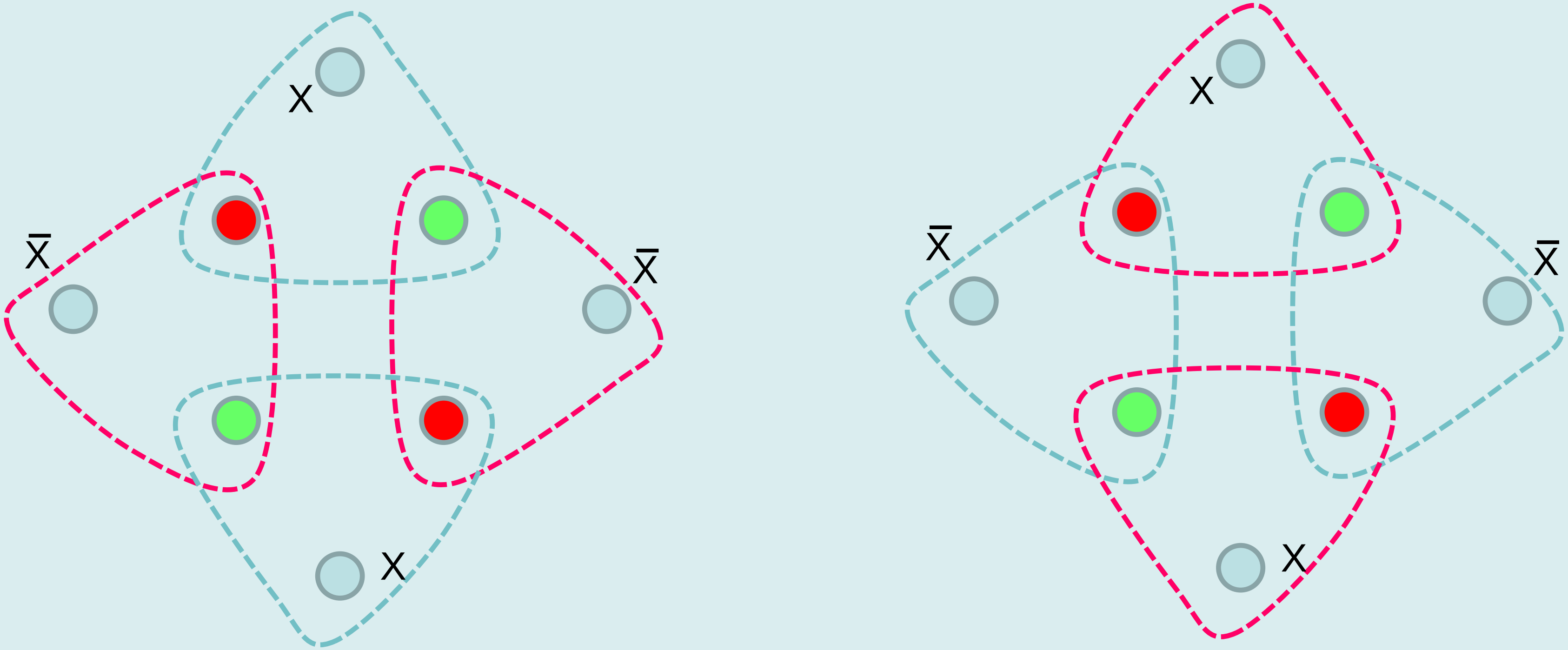


Two dimensional matching



Three dimensional matching (3DM)

3-SAT $<_P$ 3DM

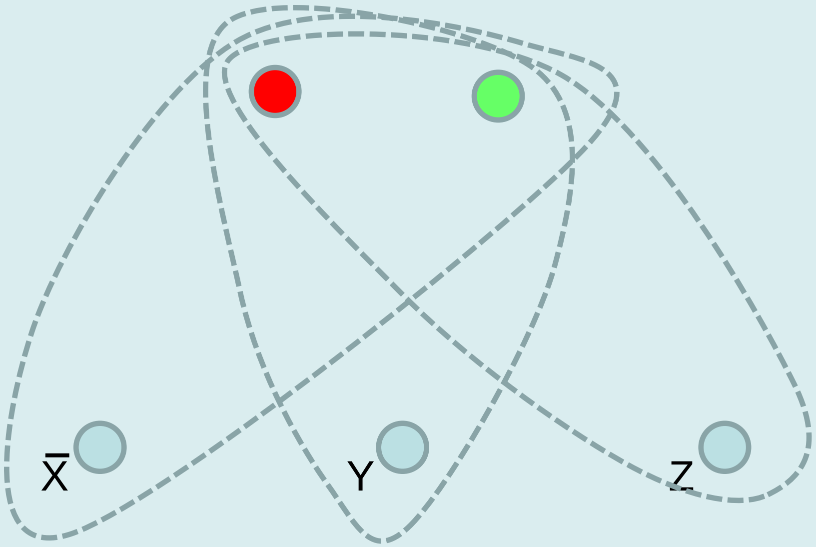


X True

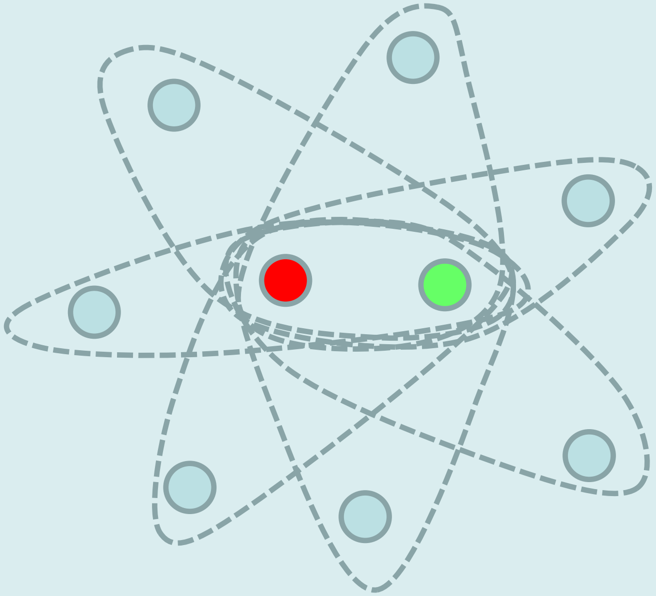
X False

Truth Setting Gadget

3-SAT $<_P$ 3DM



Clause gadget for ($\bar{X}$ OR Y OR Z)



Garbage Collection Gadget
(Many copies)

Exact Cover (sets of size 3) XC3

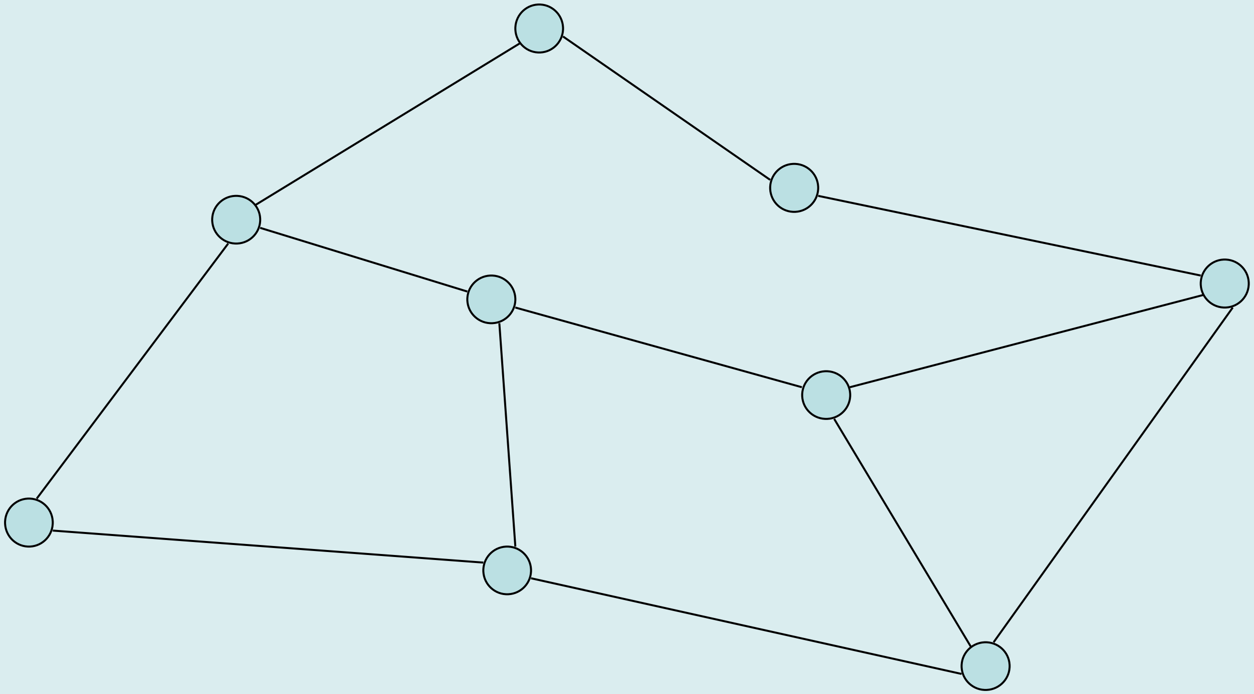
Given a collection of sets of size 3 of a domain of size 3N, is there a sub-collection of N sets that cover the sets

(A, B, C), (D, E, F), (A, B, G),
(A, C, I), (B, E, G), (A, G, I),
(B, D, F), (C, E, I), (C, D, H),
(D, G, I), (D, F, H), (E, H, I),
(F, G, H), (F, H, I)

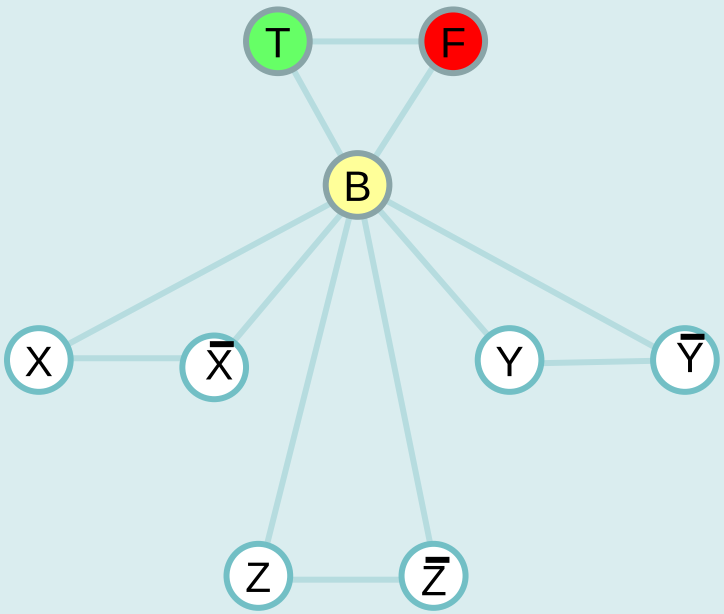
$$3DM <_P XC3$$

Graph Coloring

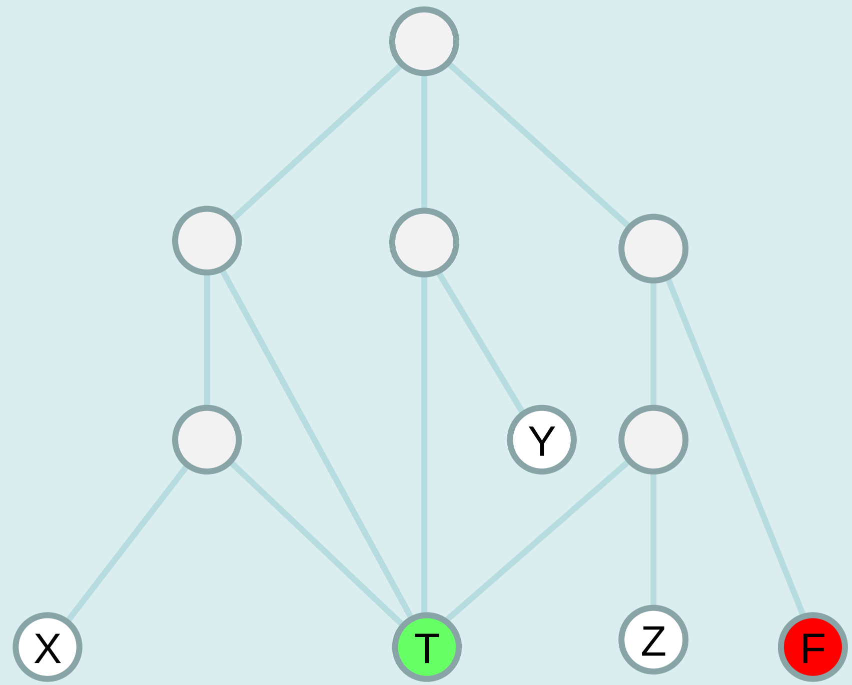
- NP-Complete
  - Graph K-coloring
  - Graph 3-coloring
- Polynomial
  - Graph 2-Coloring



3-SAT $<_P$ 3 Colorability



Truth Setting Gadget



Clause Testing Gadget

(Can be colored if at least one input is T)

Number Problems

- Subset sum problem
  - Given natural numbers w_1,. . ., w_n and a target number W, is there a subset that adds up to exactly W?

- Subset sum problem is NP-Complete
- Subset Sum problem can be solved in O(nW) time

XC3 $<_P$ SUBSET SUM

Idea: Represent each set as a large integer, where the element x_i is encoded as D^i where D is an integer

$\{x_3, x_5, x_9\} \Rightarrow D^3 + D^5 + D^9$

Does there exist a subset that sums to exactly
$D^1 + D^2 + D^3 + \ldots + D^{n-1} + D^n$

Detail: How large is D? We need to make sure that we do not have any carries, so we can choose D = m+1, where m is the number of sets.

Integer Linear Programming

- Linear Programming – maximize a linear function subject to linear constraints
- Integer Linear Programming – require an integer solution
- NP Completeness reduction from 3-SAT

Use 0-1 variables for x_i's

Constraint for clause: $(x_1 \vee \overline{x_2} \vee \overline{x_2})$

$x_1 + (1 - x_2) + (1-x_3) > 0$

Scheduling with release times and deadlines (RD-Sched)

- Tasks, $\{t_1, t_2, \ldots t_n\}$
- Task t_j has a length l_j, release time r_j and deadline d_j
- Once a task is started, it is worked on without interruption until it is completed
- Can all tasks be completed satisfying constraints?

Subset Sum $<_P$ RD-Sched

- Subset Sum Problem
  - $\{s_1, s_2, . . . , s_N\}$, integer K_1
  - Does there exist a subset that sums to K_1?
  - Assume the total sums to K_2

Reduction

- Tasks $\{t_1, t_2, . . . t_N, x\}$
- t_j has length s_j, release 0, deadline $K_2 + 1$
- x has length 1, release K_1, deadline $K_1 + 1$

Friday: NP-Completeness and Beyond!