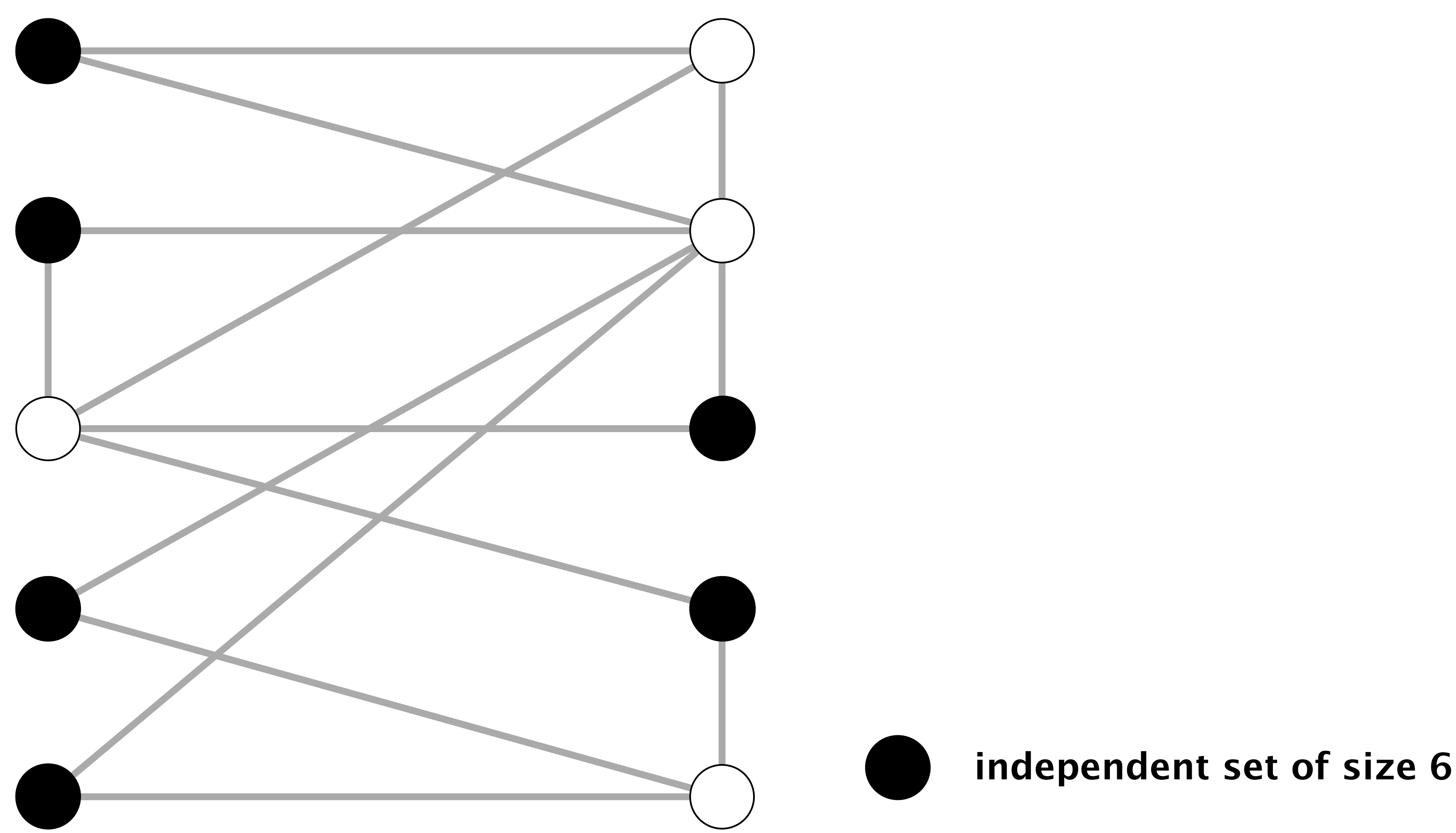
NP hardness Reductions

VERTEX-COVER $\equiv_p$ IND-SET

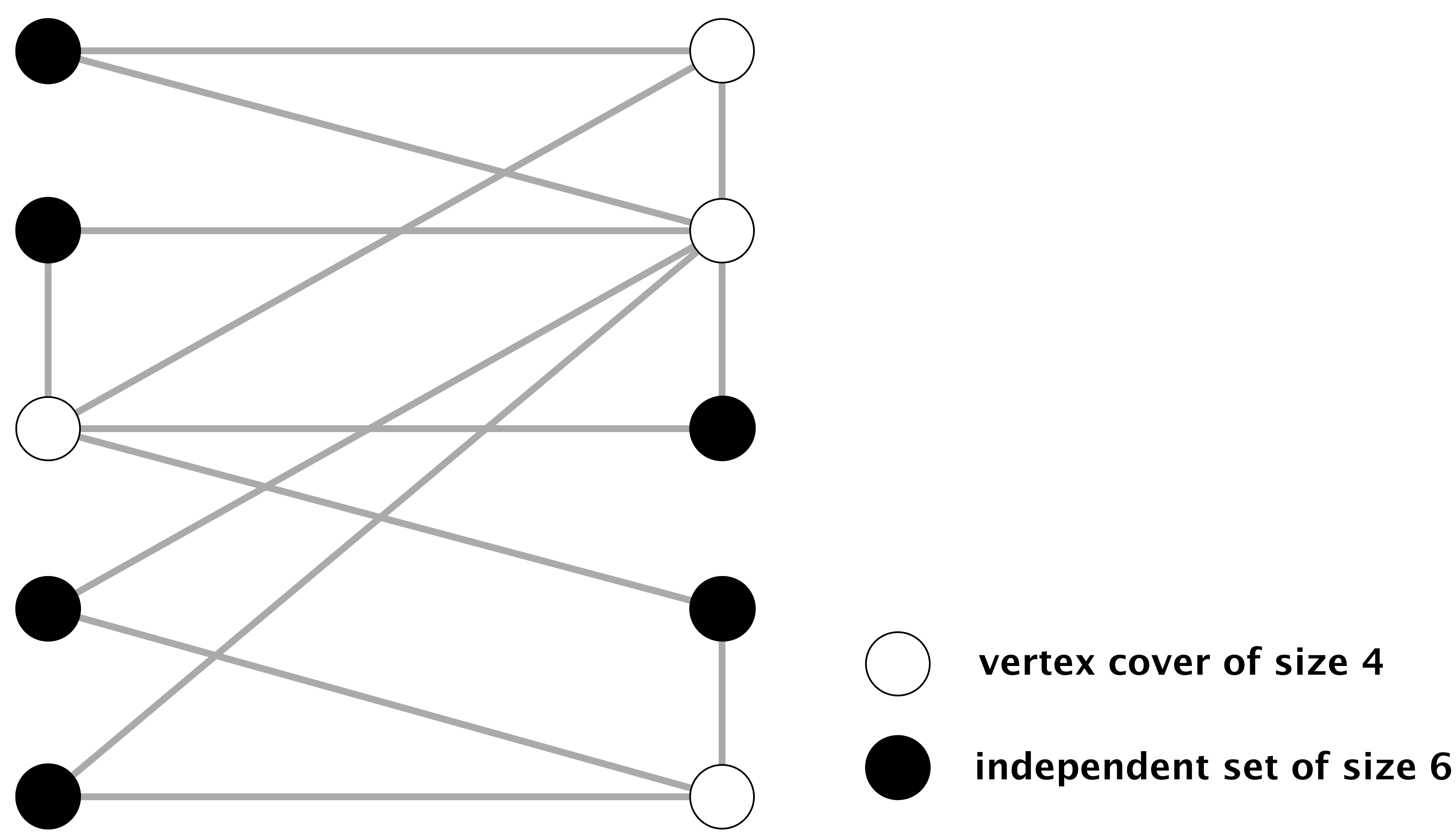
IND-SET

- Given a graph $G = (V, E)$, an independent set is a subset of vertices $S \subseteq V$ such that no two of them are adjacent, that is, for any $x, y \in S$, $(x, y) \notin E$
- **IND-SET Problem.** Given a graph $G = (V, E)$ and an integer k, does G have an independent set of size at least k?



Vertex-Cover

- Given a graph $G = (V, E)$, a vertex cover is a subset of vertices $T \subseteq V$ such that for every edge $e = (u, v) \in E$, either $u \in T$ or $v \in T$.
- **VERTEX-COVER Problem.** Given a graph $G = (V, E)$ and an integer k, does G have a vertex cover of size at most k?



Our First Reduction

- **VERTEX-COVER** $\leq_p$ **IND-SET**
  - Suppose we know how to solve independent set, can we use it to solve vertex cover?
- **Claim.** S is an independent set of size k iff $V - S$ is a vertex cover of size $n - k$.
- **Proof.** ($\Rightarrow$) Consider an edge $e = (u, v) \in E$
  - S is independent: u, v both cannot be in S
  - At least one of $u, v \in V - S$
  - $V - S$ covers e
  - $\blacksquare$

Our First Reduction

- **VERTEX-COVER** $\leq_p$ **IND-SET**
  - Suppose we know how to solve independent set, can we use it to solve vertex cover?
- **Claim.** S is an independent set of size k iff $V - S$ is a vertex cover of size $n - k$.
- **Proof.** ($\Leftarrow$) Consider an edge $e = (u, v) \in E$
  - $V - S$ is a vertex cover: at least one of u, v must be in $V - S$
  - Both u, v cannot be in S
  - Thus, S is an independent set. ∎

Vertex Cover $\equiv_p$ IND Set

- **VERTEX-COVER** $\leq_p$ **IND-SET**
- **Reduction.** Let $G' = G,\ k' = n - k$.
    - $(\Rightarrow)$ If G has a vertex cover of size at most k then G' has an independent set of size at least k'
    - $(\Leftarrow)$ If G' has an independent set of size at least k' then G has a vertex cover of size at most k
- **IND-SET** $\leq_p$ **VERTEX-COVER**
    - Same reduction works: $G' = G$, $k' = n - k$
- **VERTEX-COVER** $\equiv_p$ **IND-SET**

VERTEX-COVER $\leq_p$ SET-COVER

Set Cover

- **Set-Cover.** Given a set U of elements, a collection $\mathcal{S}$ of subsets of U and an integer k, are there **at most** k subsets $S_1, \ldots, S_k$ whose union covers U, that is, $U \subseteq \cup_{i=1}^{k} S_i$

$U = \{ 1, 2, 3, 4, 5, 6, 7 \}$

$S_a = \{ 3, 7 \}$ $\quad$ $S_b = \{ 2, 4 \}$

$S_c = \{ 3, 4, 5, 6 \}$ $\quad$ $S_d = \{ 5 \}$

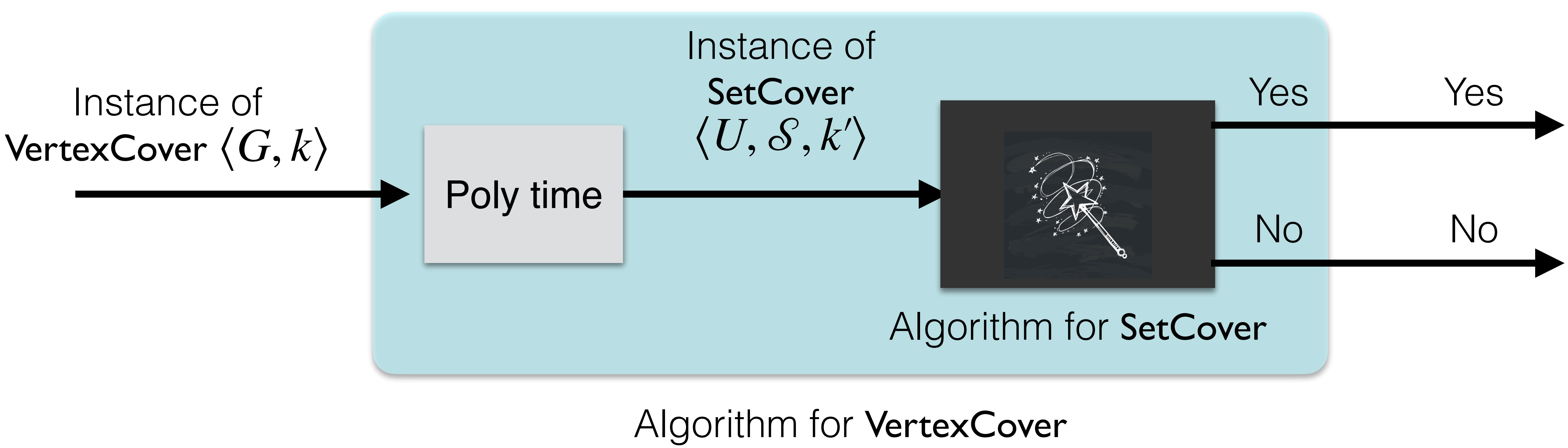
$S_e = \{ 1 \}$ $\quad$ $S_f = \{ 1, 2, 6, 7 \}$

$k = 2$

a set cover instance

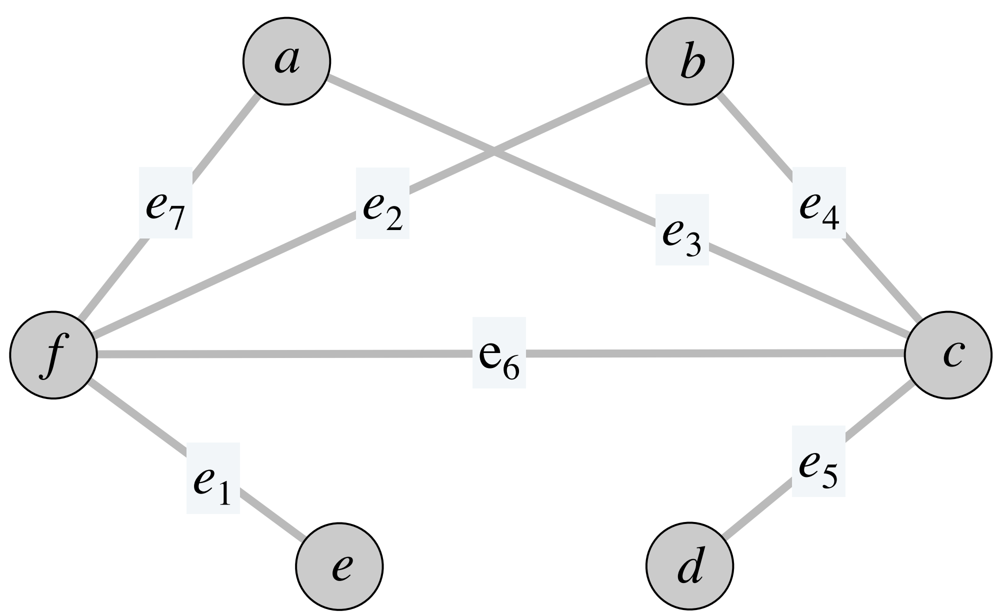
Vertex Cover $\leq_p$ Set Cover

- **Theorem.** VERTEX-COVER $\leq_p$ SET-COVER
- **Proof.** Given instance $\langle G, k \rangle$ of vertex cover, construct an instance $\langle U, \mathcal{S}, k' \rangle$ of set cover problem such that
- G has a vertex cover of size at most k if and only if $\langle U, \mathcal{S}, k' \rangle$ has a set cover of size at most k'.



Vertex Cover $\leq_p$ Set Cover

- **Theorem.** VERTEX-COVER $\leq_p$ SET-COVER
- **Proof.** Given instance $\langle G, k \rangle$ of vertex cover, construct an instance $\langle U, \mathcal{S}, k \rangle$ of set cover problem that has a set cover of size k iff G has a vertex cover of size k.
- **Reduction.** $U = E$, for each node $v \in V$, let $S_v = \{e \in E \mid e \text{ incident to } v\}$



vertex cover instance
(k = 2)

$U = \{ e_1, e_2, \ldots, e_7 \}$

$S_a = \{ e_3, e_7 \}$ $\quad S_b = \{ e_2, e_4 \}$

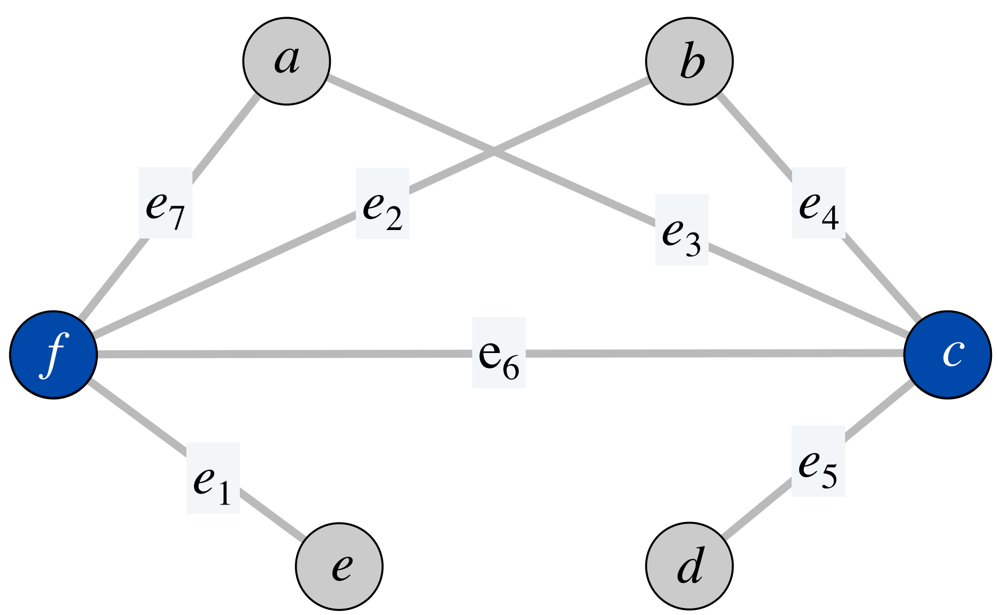
$S_c = \{ e_3, e_4, e_5, e_6 \}$ $\quad S_d = \{ e_5 \}$

$S_e = \{ e_1 \}$ $\quad S_f = \{ e_1, e_2, e_6, e_7 \}$

set cover instance
(k = 2)

Correctness

- **Claim.** $(\Rightarrow)$ If G has a vertex cover of size at most k, then U can be covered using at most k subsets.
- **Proof.** Let $X \subseteq V$ be a vertex cover in G
    - Then, $Y = \{S_v \mid v \in X\}$ is a set cover of U of the same size



vertex cover instance
(k = 2)

$U = \{ e_1, e_2, \ldots, e_7 \}$

$S_a = \{ e_3, e_7 \}$ $S_b = \{ e_2, e_4 \}$

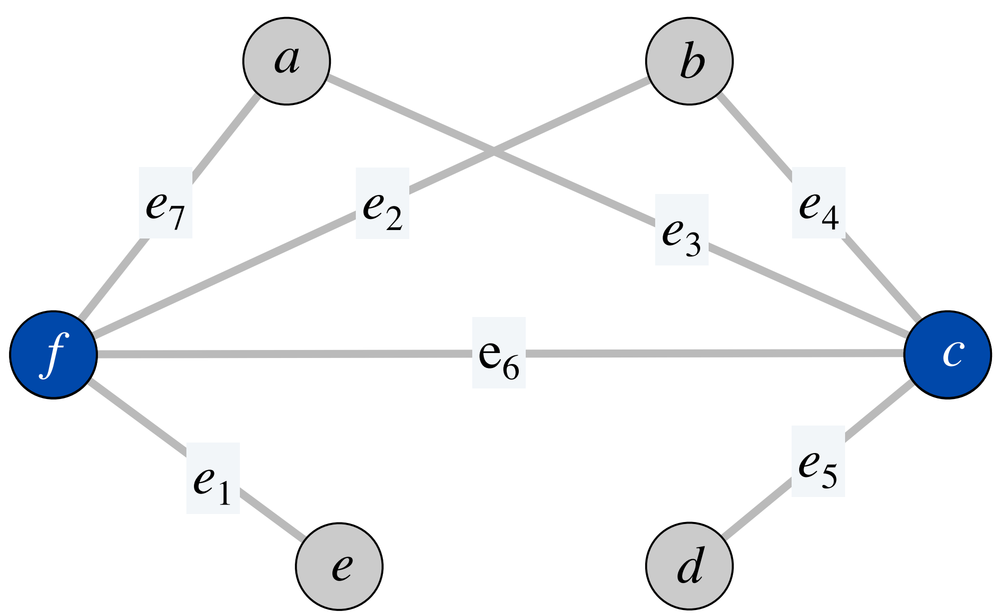
$S_c = \{ e_3, e_4, e_5, e_6 \}$ $S_d = \{ e_5 \}$

$S_e = \{ e_1 \}$ $S_f = \{ e_1, e_2, e_6, e_7 \}$

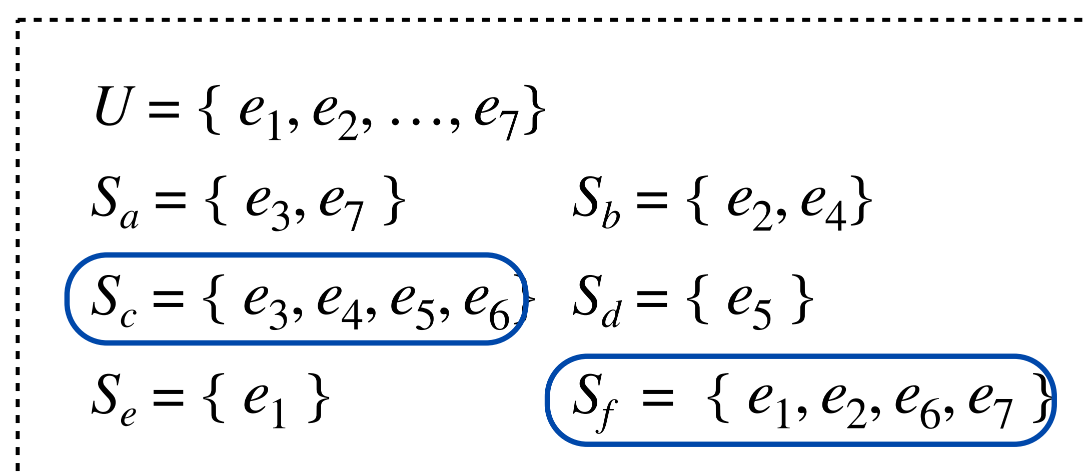
set cover instance
(k = 2)

Correctness

- **Claim.** ($\Leftarrow$) If U can be covered using at most k subsets then G has a vertex cover of size at most k.
- **Proof.** Let $Y \subseteq \mathcal{S}$ be a set cover of size k
  - Then, $X = \{v \mid S_v \in Y\}$ is a vertex cover of size k



vertex cover instance
(k = 2)



set cover instance
(k = 2)

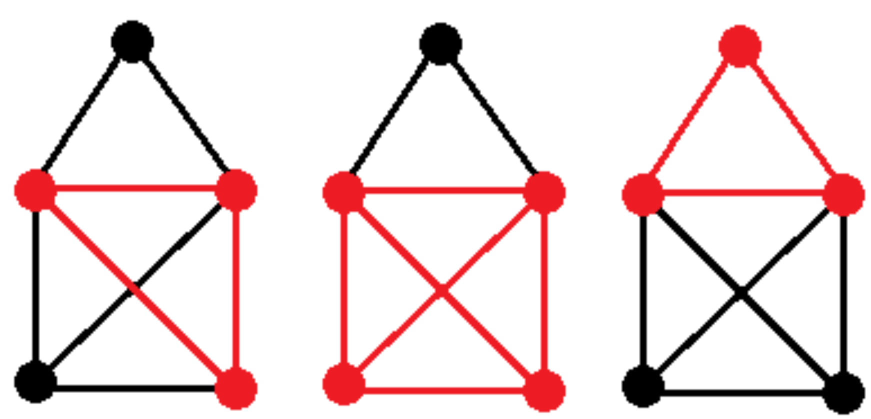
Class Exercise

IND-SET $\leq_p$ Clique

Independent set: no two vertices in the set share an edge

Clique

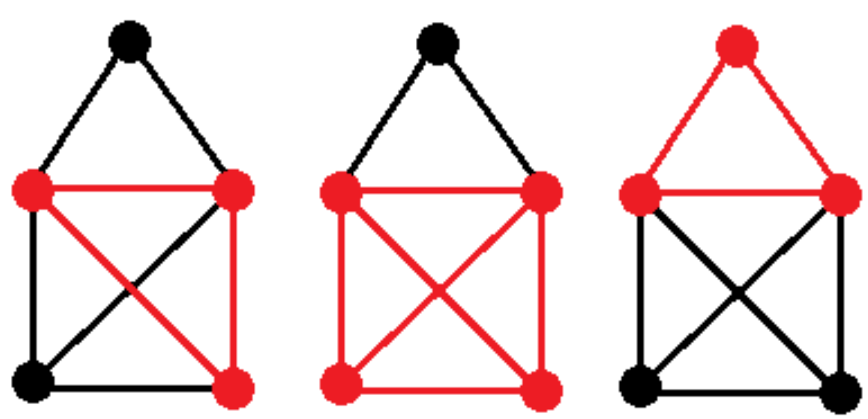
- A **clique** in an undirected graph is a subset of nodes such that every two nodes are connected by an edge. A k-clique is a clique that contains k nodes.
- **CLIQUE.** Given a graph G and a number k, does G contain a k-clique?



Clique

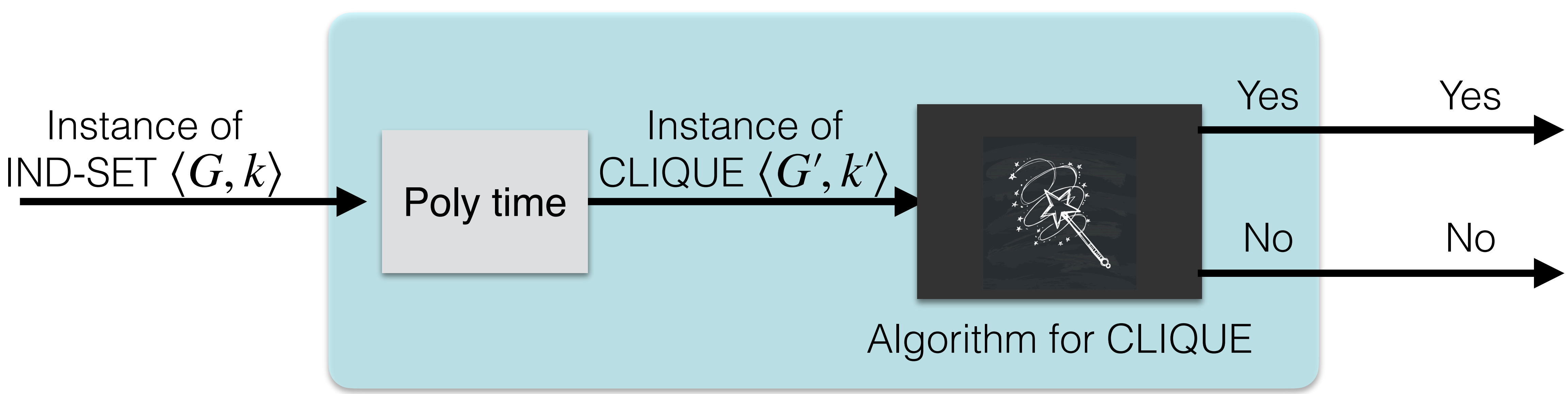
IND-SET $\leq_p$ Clique

- A **clique** in an undirected graph is a subset of nodes such that every two nodes are connected by an edge. A k-clique is a clique that contains k nodes.
- **CLIQUE.** Given a graph G and a number k, does G contain a k-clique?
- **CLIQUE** $\in$ NP
  - Certificate: a subset of vertices
  - Poly-time verifier: check is each pair of vertices have an edge between them and if size of subset is k



IND-SET to CLIQUE

- **Theorem.** IND-SET $\leq_p$ CLIQUE.
- **In class exercise.** Reduce IND-SET to Clique. Given instance $\langle G, k \rangle$ of independent set, construct an instance $\langle G', k' \rangle$ of clique such that
  - G has independent set of size k iff G' has clique of size k'.



Algorithm for IND-SET

IND-SET to CLIQUE

Independent set: no two vertices in the set share an edge

Clique: All pairs of vertices in the set share an edge

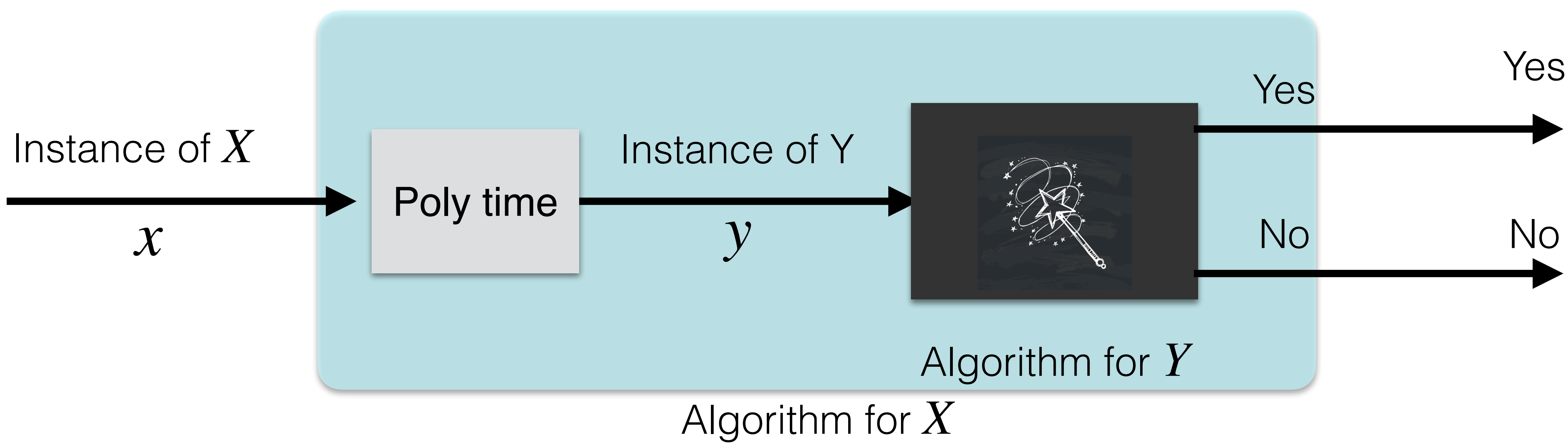
IND-SET to CLIQUE

- **Theorem.** IND-SET $\leq_p$ CLIQUE.
- Proof. Given instance $\langle G, k \rangle$ of independent set, we construct an instance $\langle G', k' \rangle$ of clique such that G has independent set of size k iff G' has clique of size k'
- **Reduction**.
  - Let $G' = (V, \overline{E})$, where $e = (u, v) \in \overline{E}$ iff $e \notin E$ and $k' = k$
  - $(\Rightarrow)$ G has an independent set S of size k, then S is a clique in G'
  - $(\Leftarrow)$ G' has a clique Q of size k, then Q is an independent set in G

Reductions: General Pattern

- Describe a polynomial-time algorithm to transform an arbitrary instance x of Problem X into a special instance y of Problem Y
- Prove that:
  - If x is a “yes” instance of X, then y is a “yes” instance of Y
  - If y is a “yes” instance of Y, then x is a “yes” instance of X

    $\iff$ if x is a "no" instance of X, then y is a "no" instance of Y



IND-SET is NP Complete:

3SAT $\leq_p$ IND-SET

Problem Definition: 3-SAT

- **Literal**. A Boolean variable or its negation x_i or $\overline{x_i}$
- **Clause**. A disjunction of literals $C_j = x_1 \vee \overline{x_2} \vee x_3$
- **Conjunctive normal form (CNF).** A boolean formula ϕ that is a conjunction of clauses $\Phi = C_1 \wedge C_2 \wedge C_3$
- **SAT**. Given a CNF formula Φ, does it have a satisfying truth assignment?
- **3SAT.** A SAT formula where each clause contains exactly 3 literals (corresponding to different variables)
- $\Phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4)$
- **SAT**, **3SAT** are both NP complete
- We will use 3SAT to prove other problems are NP hard

IND-SET

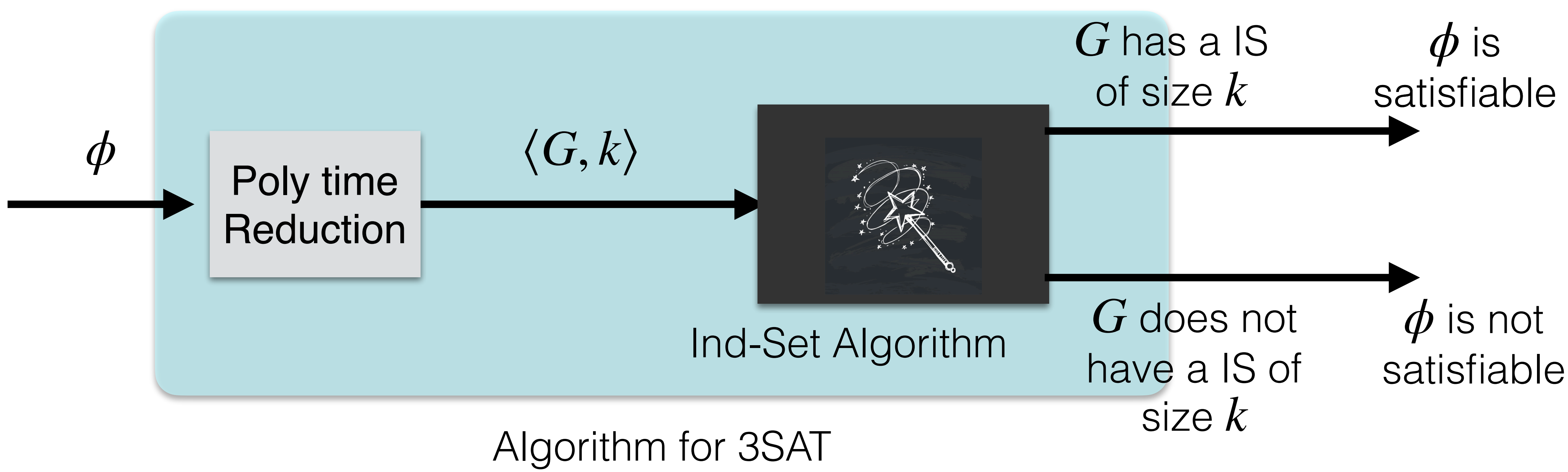
- Given a graph $G = (V, E)$, an independent set is a subset of vertices $S \subseteq V$ such that no two of them are adjacent, that is, for any $x, y \in S$, $(x, y) \notin E$
- IND-SET Problem.
  Given a graph $G = (V, E)$ and an integer k, does G have an independent set of size at least k?

IND-SET: NP Complete

- To show Independent set is NP complete
    - Show it is in NP (already did in previous lectures)
    - Reduce a known NP complete problem to it
        - We will use 3-SAT
- Looking ahead: once we have shown 3-SAT $\leq_p$ IND-SET
    - Since **IND-SET** $\leq_p$ **Vertex Cover**
    - And **Vertex Cover** $\leq_p$ **Set Cover**
    - We can conclude they are also NP hard
    - As they are both in NP, they are also NP complete!

IND-SET: NP hard

- **Theorem.** 3-SAT $\leq_p$ IND-SET
- Given an instance Φ of 3-SAT, we construct an instance $\langle G, k \rangle$ of IND-SET s.t. G has an independent set of size k iff ϕ is satisfiable.

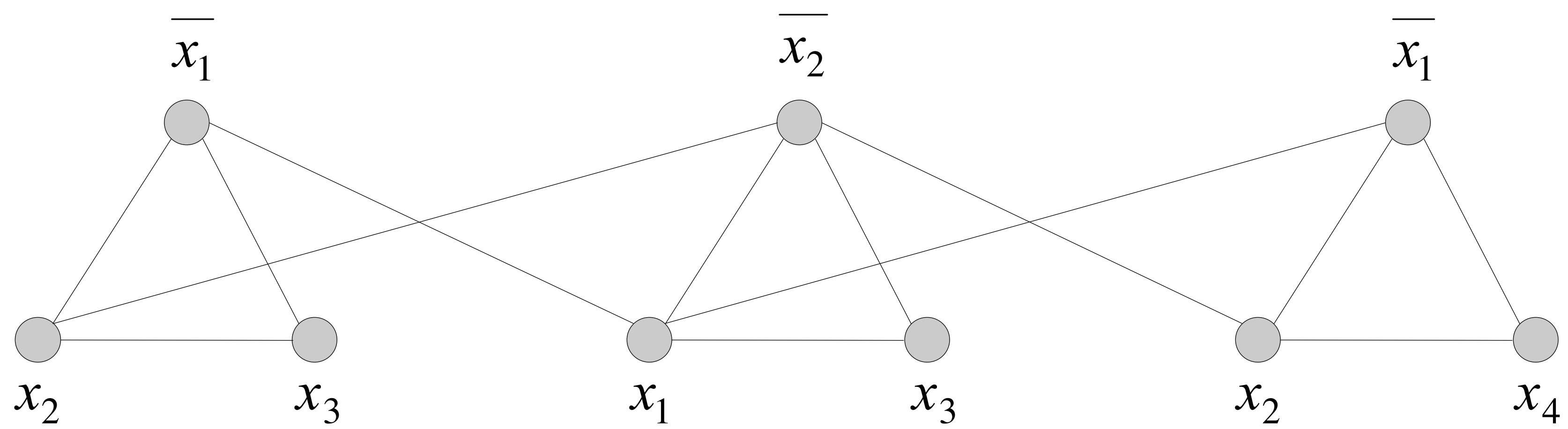


Map the Problems

3SAT		**Ind-Set**
	What is a possible solution?	
An assignment of T/F to variables		A selection of vertices to be an IS S
	What is the requirement?	
Each clause must contain at least one literal that is True		S must contain at least k vertices
	What are the restrictions?	
x can be true iff $\bar{x}$ is assigned false		If $(u, v) \in E$, then both u and v cannot be in S

$3SAT \leq_p$ IND-SET

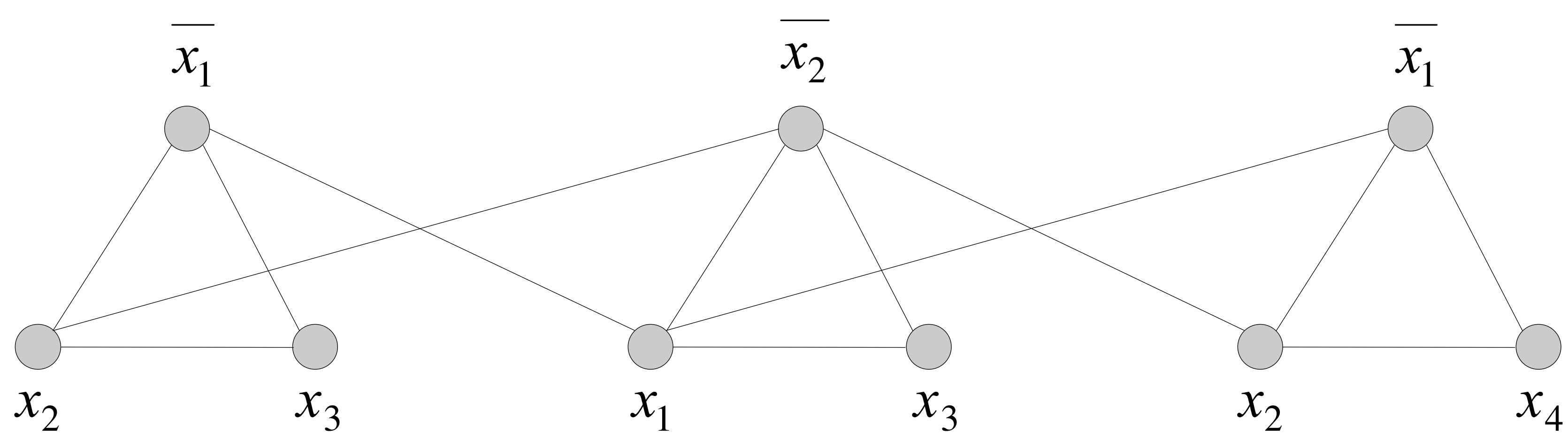
- **Reduction.** Let k be the number of clauses in Φ.
  - G has $3k$ vertices, one for each literal in Φ
  - (Clause gadget) For each clause, connect the three literals in a triangle
  - (Variable gadget) Each variable is connected to its negation in any other clause



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

3SAT$\leq_p$ IND-SET

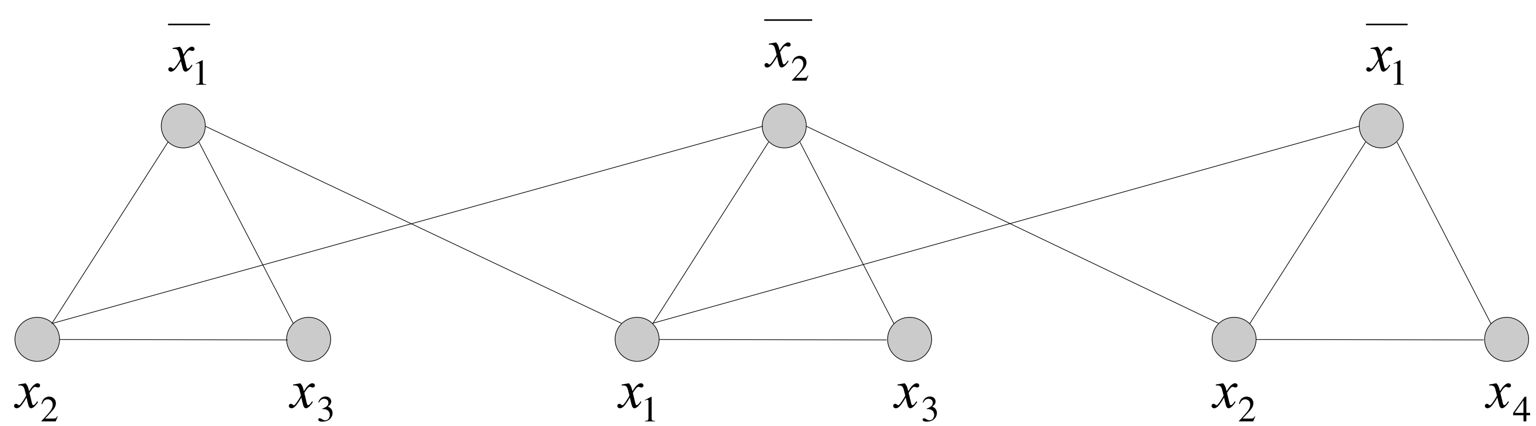
- **Observations.**
  - Any independent set is G can contain at most 1 vertex from each clause triangle
  - Only one of x_i or $\overline{x_i}$ can be in an independent set (*consistency*)



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

3SAT $\leq_p$ IND-SET

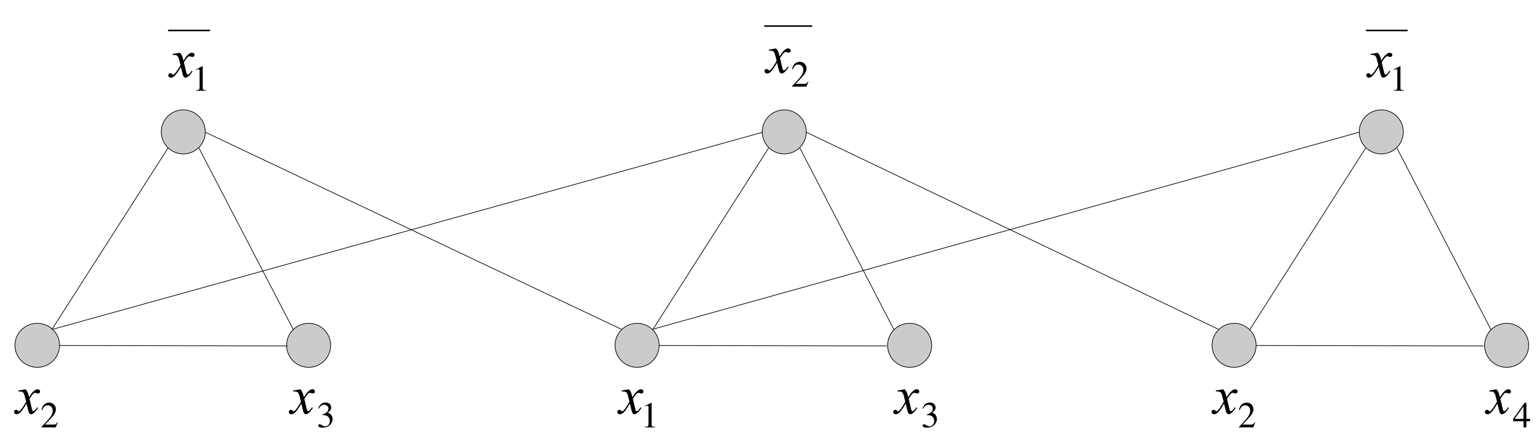
- **Claim.** Φ is satisfiable iff G has an independent set of size k
- ($\Rightarrow$) Suppose Φ is satisfiable, consider a satisfying assignment
    - There is at least one true literal in each clause
    - Select one true literal from each clause/triangle
    - This is an independent set of size k



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

3SAT$\leq_p$ IND-SET

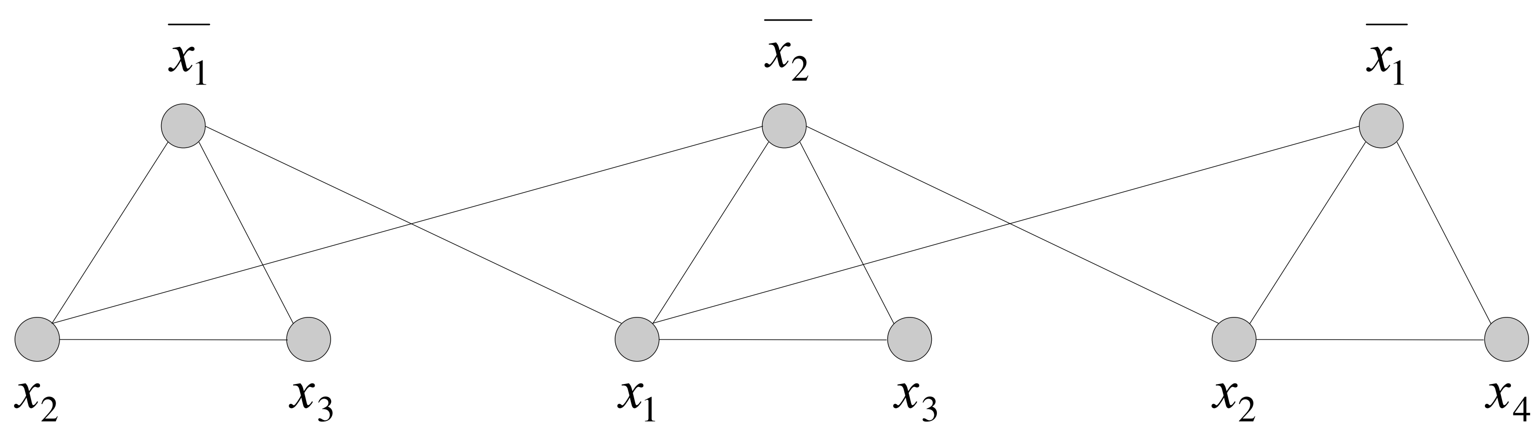
- **Claim.** Φ is satisfiable iff G has an independent set of size k
- $(\Leftarrow)$ Let S be in an independent set in G of size k
  - S must contain exactly one node in each triangle
  - Set the corresponding literals to *true*
  - Set remaining literals arbitrarily
  - All clauses are satisfied — Φ is satisfiable ∎



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

3SAT $\leq_p$ IND-SET

- Our reduction is clearly polynomial time in the input
  - G has 3k nodes, where k is #clauses, and $< (3k)^2$ edges
- Thus, independent is NP hard
- Since independent set is in NP (shown previously)
  - Independent set is NP complete



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

Reduction Strategies

- Equivalence
  - **VERTEX-COVER** $\equiv_p$ **IND-SET**
- Special case to general case
  - **VERTEX-COVER** $\leq_p$ **SET-COVER**
- Encoding with gadgets
  - **3-SAT** $\leq_p$ **IND-SET**
- Transitivity
  - **3-SAT** $\leq_p$ **IND-SET** $\leq_p$ **VERTEX-COVER** $\leq_p$ **SET-COVER**
  - Thus, **IND-SET**, **VERTEX-COVER** and **SET-COVER** are NP hard
  - Since they are all in NP, also NP - complete

SUBSET-SUM is NP Complete:

Vertex-Cover $\leq_p$ SUBSET-SUM

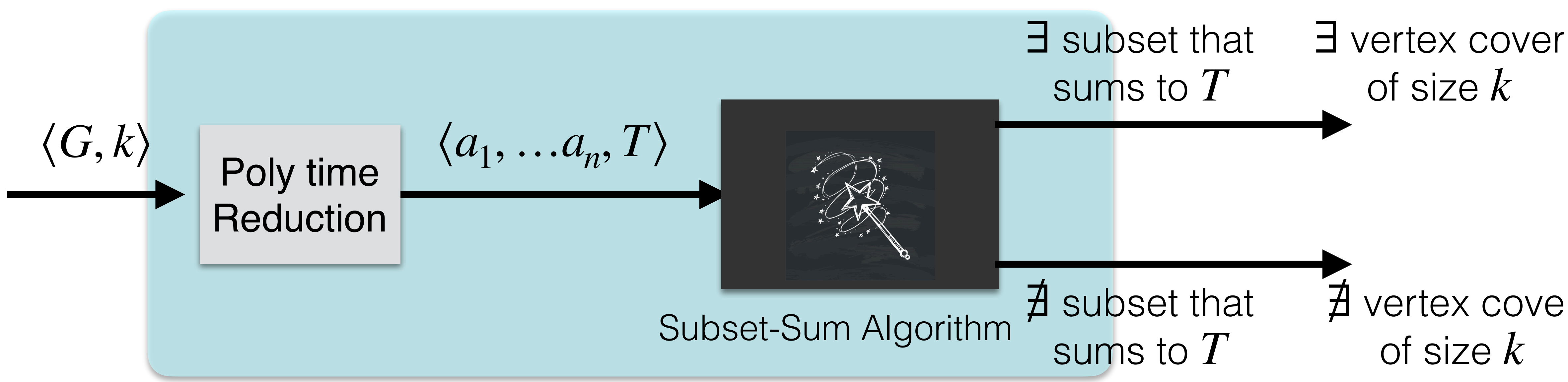
This reduction is noticeably harder than the previous ones and very clever

Subset Sum Problem

- **SUBSET-SUM.**
  Given n positive integers $a_1, \ldots, a_n$ and a target integer T, is there a subset of numbers that adds up to exactly T
- **SUBSET-SUM** $\in$ NP
  - Certificate: a subset of numbers
  - Poly-time verifier: checks if subset is from the given set and sums exactly to T
- Problem has a pseudo-polynomial $O(nT)$-time dynamic programming algorithm similar to Knapsack
- Will prove **SUBSET-SUM** is **NP hard:** reduction from vertex cover

Vertex Cover to Subset Sum

- **Theorem.** VERTEX-COVER $\leq_p$ SUBSET-SUM
- Proof. Given a graph G with n vertices and m edges and a number k, we construct a set of numbers $a_1, \ldots, a_t$ and a target sum T such that G has a vertex cover of size k iff there is a subset of numbers that sum to T



Algorithm for Vertex Cover

Map the Problems

Vertex Cover		**Subset Sum**
	What is a possible solution?	
A selection of vertices to be in VC C		A selection of numbers in subset S
	What is the requirement?	
C must contain at most k vertices		numbers in S must sum to T
	What are the restrictions?	
If $(u, v) \in E$, then either u or v must be in S		S must be a subset of input integers

Vertex Cover to Subset Sum

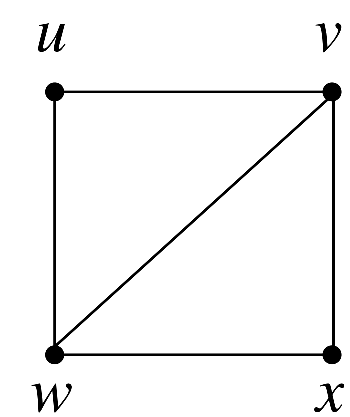
- **Theorem.** VERTEX-COVER $\leq_p$ SUBSET-SUM
- **Proof.** Label the edges of G as $0, 1, \ldots, m-1$.
- **Reduction**.
  - We'll create one integer for every vertex, and one integer for every edge
  - Force selection of k vertex integers: so will make sure that we can't sum to T unless we have that
  - Force edge covering: for every edge (u, v), we will force that number can't sum to T unless either u or v is picked

Vertex Cover to Subset Sum

- **Theorem.** VERTEX-COVER $\leq_p$ SUBSET-SUM
- Label the edges of G as $0, 1, \ldots, m-1$.
- **Reduction**. Create $n + m$ integers and a target value T as follows
- Each integer is a $m + 1$-bit number in base four
- Vertex integer a_v : mth (most significant) bit is 1 and for $i < m$, the ith bit is 1 if ith edge is incident to vertex v
- Edge integer b_{uv} : mth digit is 0 and for $i < m$, the ith bit is 1 if this integer represents an edge $i = (u, v)$
- Target value $T = k \cdot 4^m + \sum_{i=0}^{m-1} 2 \cdot 4^i$

Vertex Cover to Subset Sum

- Example: consider the graph $G = (V, E)$ where $V = \{u, v, w, x\}$ and $E = \{(u,v),\ (u,w),\ (v,w),\ (v,x),\ (w,x)\}$



	5th	4th : (wx)	3rd : (vx)	2nd : (vw)	1st : (uw)	0th: (uv)
a_u	1	0	0	0	1	1
a_v	1	0	1	1	0	1
a_w	1	1	0	1	1	0
a_x	1	1	1	0	0	0
b_{uv}	0	0	0	0	0	1
b_{uw}	0	0	0	0	1	0
b_{vw}	0	0	0	1	0	0
b_{vx}	0	0	1	0	0	0
b_{wx}	0	1	0	0	0	0

$$a_u := 111000_4 = 1344$$
$$a_v := 110110_4 = 1300$$
$$a_w := 101101_4 = 1105$$
$$a_x := 100011_4 = 1029$$

$$b_{uv} := 010000_4 = 256$$
$$b_{uw} := 001000_4 = 64$$
$$b_{vw} := 000100_4 = 16$$
$$b_{vx} := 000010_4 = 4$$
$$b_{wx} := 000001_4 = 1$$

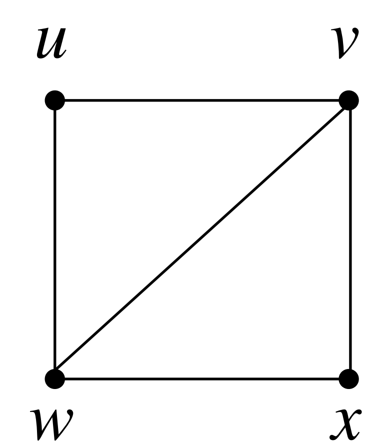
- If $k = 2$ then $T = 222222_4 = 2730$

Correctness

- **Claim.** G has a vertex cover of size k if and only there is a subset X of corresponding integers that sums to value T
- ($\Rightarrow$) Let C be a vertex cover of size k in G, define X as
  $X := \{a_v \mid v \in C\} \cup \{b_i \mid$ edge i has exactly one endpoint in $C\}$

	5^{th}	4^{th} : (wx)	3^{rd} : (vx)	2^{nd} : (vw)	1^{st} : (uw)	0^{th} : (uv)
a_u	1	0	0	0	1	1
a_v	1	0	1	1	0	1
a_w	1	1	0	1	1	0
a_x	1	1	1	0	0	0
b_{uv}	0	0	0	0	0	1
b_{uw}	0	0	0	0	1	0
b_{vw}	0	0	0	1	0	0
b_{vx}	0	0	1	0	0	0
b_{wx}	0	1	0	0	0	0

$C = \{v, w\}$



$T = 222222_4 = 2730$

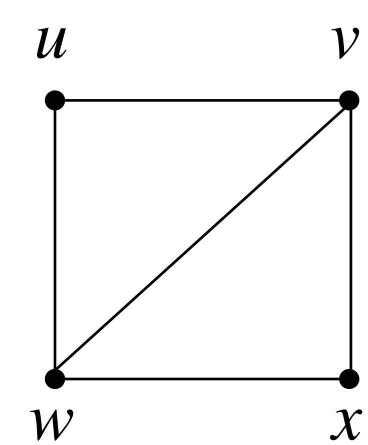
$$T = k \cdot 4^m + \sum_{i=0}^{m-1} 2 \cdot 4^i$$

Correctness

- **Claim.** G has a vertex cover of size k if and only there is a subset X of corresponding integers that sums to value T
- ($\Rightarrow$) Let C be a vertex cover of size k in G, define X as
  $X := \{a_v \mid v \in C\} \cup \{b_i \mid$ edge i has exactly one endpoint in $C\}$

	5th	4th : (wx)	3rd : (vx)	2nd : (vw)	1st : (uw)	0th: (uv)
[redacted]						
a_v	1	0	1	1	0	1
a_w	1	1	0	1	1	0
[redacted]						
b_{uv}	0	0	0	0	0	1
b_{uw}	0	0	0	0	1	0
[redacted]						
b_{vx}	0	0	1	0	0	0
b_{wx}	0	1	0	0	0	0

$$C = \{v, w\}$$



$$T = 222222_4 = 2730$$

$$T = k \cdot 4^m + \sum_{i=0}^{m-1} 2 \cdot 4^i$$

Correctness

- **Claim.** G has a vertex cover of size k if and only there is a subset X of corresponding integers that sums to value T
- $(\Rightarrow)$ Let C be a vertex cover of size k in G, define X as
  $X := \{a_v \mid v \in C\} \cup \{b_i \mid \text{edge } i \text{ has exactly one endpoint in } C\}$
- Sum of the most significant bits of X is k
- All other bit must sum to 2, why?
- Consider column for edge (u, v):
  - Either both endpoints are in C, then we get two 1's from a_v and a_u and none from b_{uv}
  - Exactly one endpoint is in C: get 1 bit from b_{uv} and 1 bit from a_u or a_v
- Thus the elements of X sum to exactly T

Vertex Cover to Subset Sum

- **Claim.** G has a vertex cover of size k if and only there is a subset X of corresponding integers that sums to value T
- $(\Leftarrow)$ Let X be the subset of numbers that sum to T
- That is, there is $V' \subseteq V, E' \subseteq E$ s.t.
$$X := \sum_{v \in V'} a_v + \sum_{i \in E'} b_i = T = k \cdot 4^m + \sum_{i=0}^{m-1} 2 \cdot 4^i$$
- These numbers are base 4 and there are no carries
- Each b_i only contributes 1 to the ith digit, which is 2
- Thus, for each edge i, at least one of its endpoints must be in V'
    - V' is a vertex cover
- Size of V' is k: only vertex-numbers have a 1 in the mth position

Subset Sum: Final Thoughts

- Polynomial time reduction?
  - $O(nm)$ since we check vertex/edge incidence for each vertex/edge when creating $n + m$ numbers
- Does a $O(nT)$ subset-sum algorithm mean vertex cover can be solved in polynomial time?
  - No! $T \approx 4^m$
- NP hard problems that have pseudo-polynomial algorithms are called *weakly NP hard*

Steps to Prove X is NP Complete

- Step 1. Show X is in **NP**
- Step 2. Pick a known NP hard problem Y from class
- Step 3. Show that $Y \leq_p X$
  - Show both sides of reduction are correct: if and only if directions
  - State that reduction runs in polynomial time in input size of problem Y

List of NPC Problems So Far

- SAT/ 3-SAT
- INDEPENDENT SET
- VERTEX COVER
- SET COVER
- CLIQUE
- 3-COLOR (k-coloring of graphs for $k \geq 3$ is also hard.)
- Subset-Sum
- Knapsack
- Next:
    - Traveling salesman problem
    - Hamiltonian cycle / path