$SAT \leq_P 3SAT$

Claim: $SAT \leq_P 3SAT$

Proof: Convert SAT clauses with > 3 literals into $3SAT$ clauses.

$$\phi_{SAT} = (x_1 \vee x_2 \vee x_3 \vee \cdots \vee x_k)$$
$$\rightarrow \phi_{3SAT} = (x_1 \vee x_2 \vee \mathbf{z_1}) \wedge (\overline{\mathbf{z_1}} \vee x_3 \vee \mathbf{z_2}) \wedge \cdots \wedge (\overline{\mathbf{z_{k-3}}} \vee x_{k-1} \vee x_k)$$

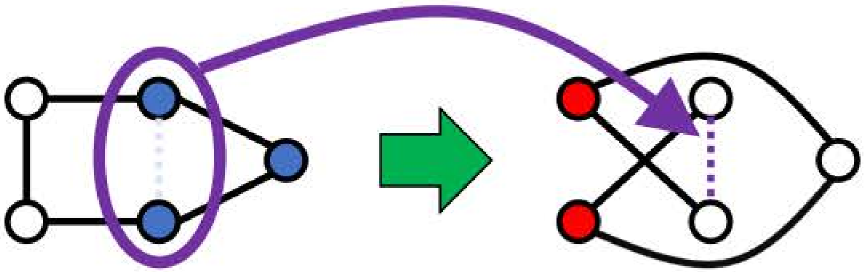
Need to show: ϕ_{SAT} can be true $\Leftrightarrow$ ϕ_{3SAT} can be true.

Suppose ϕ_{SAT} can be true. Then some x_m is true. Let x_m be true in ϕ_{3SAT}. Let all z_i's before x_m be true and all z_i's after be false.
$\Rightarrow$ Every clause has a variable set to true. $\therefore \phi_{3SAT} = T$.

Suppose ϕ_{3SAT} can be true. Some x_m must be true. If not, all z_i's must be true, and last clause would be false. $\therefore \phi_{SAT} = T$.

$\therefore SAT \leq_P 3SAT$

Vertex Cover (VC)



Claim: $CLIQUE \leq_P$VC

Proof: Let $G = (V, E)$, k be input to the clique problem, where $|V| = n$. Construct the complement graph $\bar{G} = (V, \bar{E})$ by checking each pair of vertices and making them an edge in $\bar{E}$ if they are not an edge in E. $O(n^2)$ time

$$G \text{ has a } k\text{-clique} \Leftrightarrow \bar{G} \text{ has an } (n-k)\text{-VC}.$$

$\Rightarrow$ Suppose G has a k-clique Q. Consider $C = V\backslash$Q. For C to be a VC of $\bar{G}$, each edge in $\bar{E}$ must contain a vertex from C. Consider an edge e in $\bar{E}$ where neither vertex is in C…

$\Leftarrow$ Suppose $\bar{G}$ has an $\boldsymbol{(n-k)}$**-VC** $\boldsymbol{C}$. Consider $\boldsymbol{Q = V\backslash C}$. For Q to be a clique in G, each pair of its vertices must share an edge in E. Consider a **pair of vertices** in Q that do not share an edge in E…

Vertex Cover (VC)

Vertex Cover: Given graph $G = (V, E)$ and integer $k \leq |V|$, is there $V' \subseteq V$, with $|V'| \leq k$, such that each edge in E contains an end point in V'?

Claim: VC $\in NP$

Proof:

Build a polynomial time verifier.

$|V| = n.$

M = on input $\langle\langle G, k\rangle, V'\rangle$, where V' is a subset of V.

$\boldsymbol{O(1)} \longrightarrow$ 1. Test if $|V'| \leq k$, reject if not.

$\boldsymbol{O(n^2)} \longrightarrow$ 2. For each edge $e = (a, b)$ in E,

$\boldsymbol{O(n)} \longrightarrow$ 2.1 Test if $a \in V'$ or $b \in V'$, reject if neither.

$\boldsymbol{O(1)} \longrightarrow$ 3. accept.

For $|V| = n$, M runs in $O(n^3)$ time, therefore $VC \in NP$.

1. For the graph $G = C_3$ with $V(C_3) = \{x_1, x_2, x_3\}$, $|C| = 2$ and $|I| = 1$ for all the cases because $C = \{x_1, x_2\}$ or $\{x_1, x_3\}$ or $\{x_2, x_3\}$; and $I = \{x_1\}$ or $\{x_2\}$ or $\{x_3\}$. So the argument is FALSE because every graph G does not satisfy the given equality.

2. I think this is TRUE because of the definitions. Since C includes at least one end point of each edge, $V(G) - C$ must include either none of the endpoints of one edge or 1 of them. So it is an independent set by definition.

3. For this one, I can give a counterexample. Consider the graph $G(V, E)$ with $V = \{x_0, x_1, x_2, x_3\}$ and $E = \{\{x_0, x_1\}, \{x_0, x_2\}, \{x_0, x_3\}\}$ that is a minimal example of a tree(but also a graph) with 3 leaves as shown here. In this tree, notice the set $\{x_0\}$ is one of the vertex covers. So we can find a vertex cover C with $|C| = 1$, where $|C| = 1 < 3/2 = |E(G)/2|$. So the argument is FALSE because every vertex cover of given graph G does not satisfy the given inequality.

4. Same counterexample in part 3 can be given for this case. Only difference is the last inequality, which is for this case, $|C| = 1 < 2 = |V(G)/2|$.

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

   Given a graph $G = (V, E)$, where $|V| = n$, and a subset $S \subseteq V$, where $|S| \geq k$, check if all pairs of vertices in S are in E. Running time: $O(n^2)$.

2. $3SAT \leq_P CLIQUE$

$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

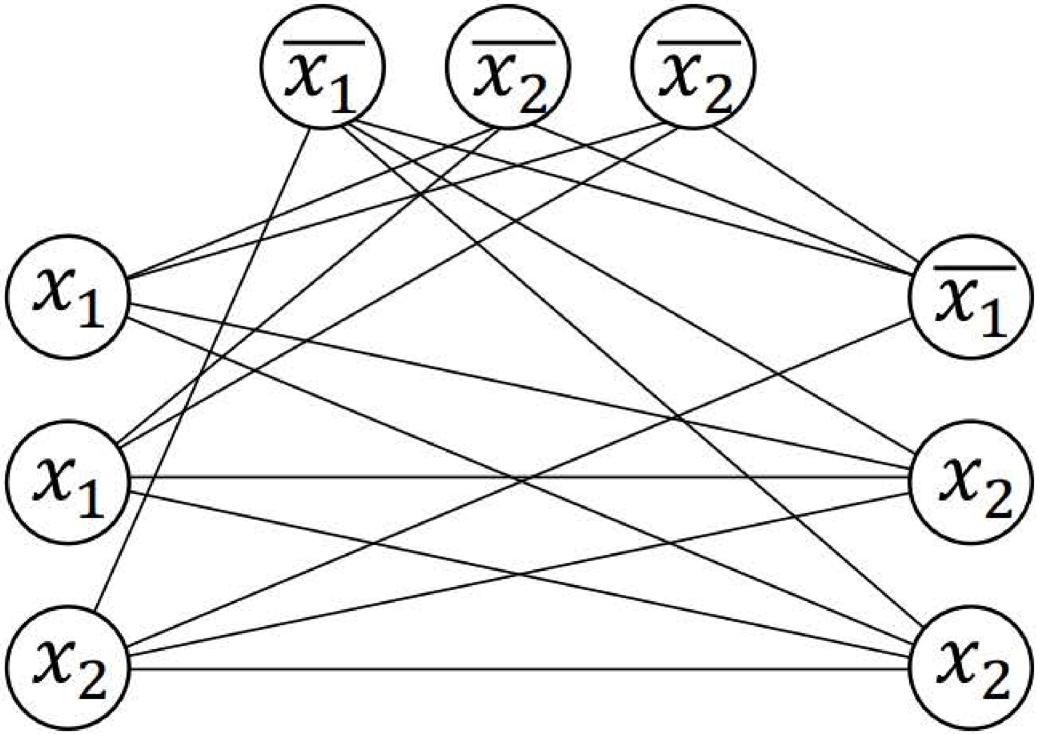
Proof: **ϕ is satisfiable $\Leftrightarrow$ G has a k-clique.**

$\Rightarrow$ Suppose ϕ is satisfiable. Then at least one literal is true in each clause. For each clause, select a node in G for one of the true literals. This forms a k-clique, since k nodes are selected and each is joined by an edge.

$\Leftarrow$ Suppose G has a k-clique. Then there is a node from the k-clique in each clause. Making each node in the k-clique true results in ϕ being true.

For each clause in ϕ, make a node for each literal. Make edge between every pair of nodes, except:

1. Nodes in the same clause
2. Nodes that are negations of each other.



$CLOSE - PAIR$

Claim: $CLOSE - PAIR = \{\langle S, d\rangle : S$ is a set of points in $\mathbb{R}^2$ with some pair less than d apart$\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|S| = n$.

M = on input $\langle S, d\rangle$

$\boldsymbol{O(n^2)} \longrightarrow$ 1. Repeat for each pair of points (a, b) in S:

$\boldsymbol{O(1)} \longrightarrow$ 2. Find distance between a and b.

$\boldsymbol{O(1)} \longrightarrow$ 3. If closest pair < d, <u>accept</u>. Otherwise, <u>reject</u>.

M is a decider and runs in $O(n^2)$ time,
therefore $CLOSE - PAIR \in P$.

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

   Given a graph $G = (V, E)$, where $|V| = n$, and a subset $S \subseteq V$, where $|S| \geq k$, check if all pairs of vertices in S are in E. Running time: $O(n^2)$.

2. ???

Independent Set (IS)

Independent Set: Given a graph $G = (V, E)$ an
integer $k \leq |V|$, is there $V' \subseteq V$, with $|V'| \geq k$
such that no two vertices $\in V'$ are adjacent?

Claim: IS $\in NP$

Proof:

Build a polynomial time verifier.

$|V| = n.$

M = on input $\langle\langle G, k\rangle, V'\rangle$, where V' is a subset of V.

$\boldsymbol{O(1)} \longrightarrow$ 1. Test if $|V'| \geq k$, <u>reject</u> if not.

$\boldsymbol{O(n^2)} \longrightarrow$ 2. For each pair of vertices v_1, v_2 in V',

$\boldsymbol{O(n^2)} \longrightarrow$ 2.1 Test if $(v_1, v_2) \in E$ and <u>reject</u> if it is.

$\boldsymbol{O(1)} \longrightarrow$ 3. <u>accept</u>.

For $|V| = n$, M runs in $O(n^4)$ time, therefore IS $\in NP$.

NP-Complete

How to show something (B) is in NP-Complete?

1. Show it is in NP.
2. Pick some known NP-Complete problem A.
3. Show that a solver for B can solve A in polynomial extra time.

B is in NP-Complete if it satisfies two conditions:

1. $B \in NP$.
2. For some $A \in NP$-C, $A \leq_P B$.

$PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from s to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

N = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)} \longrightarrow$ 1. Mark s.

$\boldsymbol{O(n)} \longrightarrow$ 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)} \longrightarrow$ 3. For each $e = (a, b) \in E$, if a is marked, mark b.

$\boldsymbol{O(1)} \longrightarrow$ 4. If t is marked, <u>accept</u>. Otherwise, <u>reject</u>.

N is a decider and runs in $O(n^3)$ time, therefore $PATH \in P$.

CSCI 338
Homework 8

Assigned 11/1/2022, due by start of class (3:05 pm) on 11/8/2022. Please submit this assignment to the appropriate dropbox on D2L. You must follow the collaboration policy detailed on the course website.

Problem 1 (5 points). Suppose that $L_1 \in P$ and $L_2 \in P$. Prove that $L_1 \cup L_2 \in P$.

Solution. $L_1 \in P$ means there exists a polynomial time decider M_1, which runs in $O(n^{k_1})$ time. Likewise, $L_2 \in P$ means there exists a polynomial time decider M_2, which runs in $O(n^{k_2})$ time. Construct the following decider for $L_1 \cup L_2$:

$M =$ on input ω.

1. Run M_1 on ω and <u>accept</u> if M_1 does.
2. Run M_2 on ω and <u>accept</u> of M_2 does.
3. <u>reject</u>.

The running time for M will be in $O(n^{k_1} + n^{k_2}) \in O(n^{\max(k_1,k_2)})$, which is polynomial. Thus, M is a polynomial time decider for $L_1 \cup L_2$, so $L_1 \cup L_2 \in P$. □

Problem 2 (5 points). A 3-clique in a graph is a set of three vertices with an edge between each of them (e.g. a triangle). Consider the problem $3 - CLIQUE = \{\langle G \rangle :$ G is an undirected graph with a $3-$clique in it$\}$. Show that $3 - CLIQUE \in P$.

Solution. $M =$ on input $\langle G \rangle$.

1. For each grouping of three vertices v_1, v_2, and v_3:
2.   If (v_1, v_2), (v_1, v_3), and (v_2, v_3) are edges in G, <u>accept</u>.
3. <u>reject</u>.

If G has a 3-clique in it, it will be found when M considers that grouping of three vertices. Therfore, M decides $3 - CLIQUE$. The running time of M is driven by the number of groupings of three vertices. Specifically, if the total number of vertices is n, the number of groups of three is:

$$\binom{n}{3} = \frac{n!}{3!(n-3)!} = \frac{n(n-1)(n-2)(n-3)!}{3!(n-3)!} = \frac{n(n-1)(n-2)}{6} \in O(n^3)$$

So, step two of M will only be run $O(n^3)$ times, at a constant cost each time (looking for three edges). Therefore, the running time of $M \in O(n^3)$ and $3 - CLIQUE \in P$. □

Problem 3 (5 points). Graphs G and H are isomorphic if the nodes of G can be renamed so that $G = H$. Consider the problem $ISO = \{\langle G, H \rangle : G \text{ and } H \text{ are isomorphic graphs}\}$. Show that $ISO \in NP$.

Solution. $M =$ on input $\langle \langle G, H \rangle, R \rangle$, where $R : V_G \to V_H$ is a renaming of G's vertices into H's vertex names.

1. If G and H do not have same number of vertices, reject.

2. For each edge (v_1, v_2) in G, mark $(R(v_1), R(v_2)) \in H$. If it does not exist, reject.

3. Scan edges in H. If any unmarked edges, reject. Otherwise, accept.

Let $n = \max(|V_G|, |V_H|)$. Step 1 happens in $O(n)$ time. Step 2 happens in $O(n^2)$ time (maximum number of edges possible in G). Likewise, step 3 happens in $O(n^2)$ time as well. Therefore, M runs in polynomial time, and $ISO \in NP$ $\square$

Problem 4 (5 points). Prove that $P \subseteq NP$.

Solution. This result is the consequence of the definition(s) of NP. If $L \in P$, that means there exists a deterministic, polynomial time decider H for L. There are now two ways to proceed:

1. Verifying L given some "evidence" can be done by ignoring the evidence and directly solving L with H (in polynomial time). Therefore, $L \in NP$.

2. Since deterministic TMs are a subset of non-deterministic TMs, H is a non-deterministic decider for L. Therefore, $L \in NP$.

$\square$

CSCI 338
Homework 9

Assigned 11/10/2022, due by start of class (3:05 pm) on 11/15/2022. Please submit this assignment to the appropriate dropbox on D2L. You must follow the collaboration policy detailed on the course website.

Problem 1 (5 points). Prove or disprove that the following Boolean formula is satisfiable: $(x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$

Solution. There are four options for variable value assignments:

1. $x = T$, $y = T$. Then clause 4 fails.
2. $x = T$, $y = F$. Then clause 3 fails.
3. $x = F$, $y = T$. Then clause 2 fails.
4. $x = F$, $y = F$. Then clause 1 fails.

Since every possible value assignment causes a clause to fail, the formula is not satisfiable. □

Problem 2 (5 points). Show that $3 - SAT \in NP$.

Solution. $M =$ on input $\langle\langle\phi\rangle, \vec{\boldsymbol{x}}\rangle$, where $\vec{\boldsymbol{x}}$ is an assignment of truth values to the variables.

1. Evaluate ϕ.
2. If ϕ is true, accept. Otherwise, reject.

For n clauses, M runs in $O(n)$ time, therefore $3 - SAT \in NP$. □

Problem 3 (5 points). Show that the following problem is in NP: Given a set of numbers S and target value t, is there some $S' \subset S$ such that $\sum_{x \in S'} x = t$?

Solution. $M =$ on input $\langle\langle S, t\rangle, S'\rangle$, where S' is a subset of S.

1. Sum all values in S'.
2. If the sum equals t, accept. Otherwise, reject.

For $|S| = n$, M runs in $O(n)$ time, therefore the problem is in NP. $\square$

Problem 4 (5 points). Show that the following problem is in NP: Given a directed graph G, is there a directed circuit (path that starts and ends on the same node - i.e., a loop through the graph) that goes through each node exactly once?

Solution. $M =$ on input $\langle\langle G = (V, E)\rangle, L\rangle$, where L is an ordered list of vertices from G.

1. Confirm that $|L| = |V|$, <u>reject</u> if not.
2. Confirm that L contains no repeats, <u>reject</u> if not.
3. For each consecutive elements of L, confirm that they form an edge in E, <u>reject</u> if not.
4. <u>accept</u>

If $|V| = n$, the running time of step 1 is $O(1)$, step 2 is $O(n)$, step 3 is $O(n)$ if E is represented as a hash set, and step 4 is $O(1)$. Thus, M runs in $O(n)$ time and the problem is in NP. $\square$

CSCI 338
Homework 10

Assigned 11/15/2022, due by start of class (3:05 pm) on 11/29/2022. Please submit this assignment to the appropriate dropbox on D2L. You must follow the collaboration policy detailed on the course website.

Problem 1 (10 points). The Hamiltonian Cycle problem is: Given a directed graph G, is there a directed circuit (path that starts and ends on the same node - i.e., a loop through the graph) that goes through each node exactly once?

The Traveling Salesman problem is: Given a directed graph G with edge weights and a number k, is there a directed circuit (path that starts and ends on the same node - i.e., a loop through the graph) that goes through each node exactly once, and whose total cost (sum of edge weights) is at most k?

Assuming the only problem you know to be NP-Complete is the Hamiltonian Cycle problem, show that the Traveling Salesman problem is NP-Complete.

Solution. First, show TSP $\in NP$:

$M =$ on input $\langle G, L \rangle$, where L is the ordered list of vertices visited.

1. Scan L to confirm that all $v \in G$ are in L and there are no repeats (reject if it fails).
2. Scan L to confirm each sequential pair corresponds to an edge in G (reject if it fails).
3. Add up the edges formed by L and reject if sum $> k$.
4. accept.

Since this will verify solutions to TSP in polynomial time ($O(n)$ where $n =$ number of vertices), TSP $\in NP$.

Convert an instance to HC, G, into a TSP instance, G', as follows: Keep the exact same vertex and edge set. Assign weight of 1 to each edge. Make $k =$ number of vertices. This can be done in polynomial time, since it is the same size as G.

Need to show there is a HC in $G \iff$ there is a TSP of size k in G'.

Suppose there is a HC solution in G. This means there is a cycle that starts and ends at the same vertex and visits each vertex exactly once. The cost of this cycle is the number of vertices, thus it is a TSP of size k.

Suppose there is a TSP solution in G' of size at most k. This means there is a cycle that starts and ends at the same vertex and visits each vertex exactly once. Thus, it is a HC solution.

Therefore, TSP is NP-Complete. $\square$

Problem 2 (10 points). Consider the problem of 4SAT: Given a cnf Boolean formula with exactly four literals per clause, can the literals be assigned values so that the formula evaluates to true? Using 3SAT, show that 4SAT is in $NP - Complete$.

Solution. Construct a verifier for 4SAT:

$M =$ on input $\langle\langle\phi\rangle, x\rangle$, where x is an assignment of variables.

1. For each clause, if the clause evaluates to false, reject.
2. accept.

The running time of M is in $O(n)$, where n is the number of clauses since step 1 happens n times, with a constant amount of work each time. Therefore, 4SAT $\in NP$.

Show 3SAT $\leq_P$ 4SAT:

Suppose ϕ_{3SAT} is an instance of 3SAT. Turn it into a ϕ_{4SAT} instance by converting each 3SAT clause to a 4SAT clause by repeating one of the variables in the clause:

$(x_1 \vee x_2 \vee x_3) \rightarrow (x_1 \vee x_2 \vee x_3 \vee x_3)$.

Need to show ϕ_{3SAT} is satisfiable $\iff$ ϕ_{4SAT} is satisfiable:

Suppose ϕ_{3SAT} is satisfiable. Then for each clause, one of the literals would have been set to true. Then the corresponding 4SAT clause would also be true, since ORing a true or false value to a true statement stays true.

Suppose ϕ_{4SAT} is satisfiable. Then for each clause, one of the literals would also have been set to true. If the new (repeated) literal is the one set to true, then the original copy of that variable would also be true, so the corresponding 3SAT clause would also be true. If the literal set to true was not the new (repeated) one, then removing the new variable does not change the truth assignment in the corresponding 3SAT clause.

Therefore, 4SAT is in $NP - Complete$.

□

Problem 3 (10 points). Assuming the only problem you know to be NP-Complete is the 3SAT problem, show that the CLIQUE problem is NP-Complete.

Solution. Done in class on 11/17. □

Problem 4 (5 points). For a complete graph over n vertices, what is the size of the smallest Vertex Cover, Clique, and largest Independent Set?

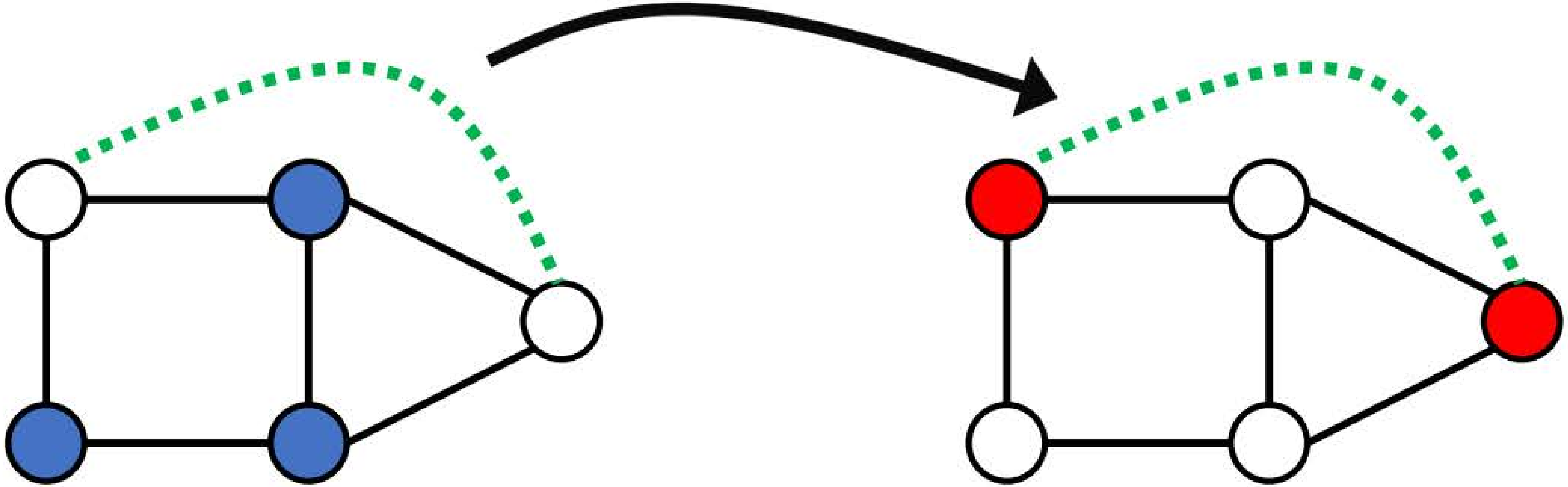
Solution. Vertex Cover $= n - 1$

Clique $=$ 0 or 1, depending on how you look at it. n if you are looking for the largest.

Independent Set $= 1$ □

Vertex Cover vs Independent Set

Prove that the compliment of a VC is an IS and that the compliment of an IS is a VC.



k - VC $\Longleftrightarrow$ $(n-k)$ - IS