NP Complete Problems

POLYNOMIAL TIME REDUCIBILITY

- Consider decision problem A, which we would like to solve in polynomial time.
- Suppose that we already know how to solve a different decision problem B in polynomial time.
- Finally, suppose that we have a procedure that transforms any instance **α** of A into some instance **β** of B with the following characteristics:

  (1) The transformation takes polynomial time.

  (2) The answers are the same. That is, the answer for **α** is "yes" if and only if the answer for **β** is also "yes."
- Such a procedure is called a **polynomial-time reduction algorithm.**

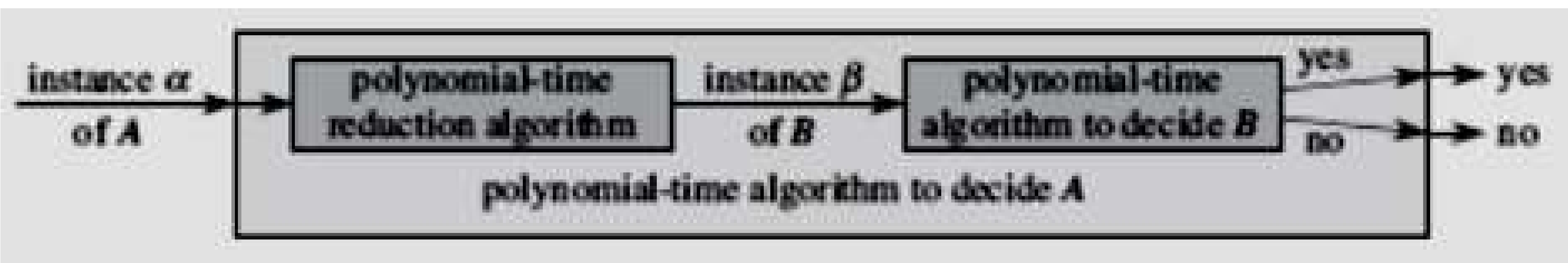
POLYNOMIAL TIME REDUCIBILITY

- It provides us a way to solve problem A in polynomial time:

  1. Given an instance **α** of problem A, use a polynomial-time reduction algorithm to transform it to an instance **β** of problem B.

  2. Run the polynomial-time decision algorithm for B on the instance **β**.

  3. Use the answer for **α** as the answer for β.



NP Complete Classes

- If a language L1 is polynomial time reducible to another language L2, then it is denoted as L1 $\leq_p$ L2.
- A language L is NP-complete if
  1. L Є NP, and
  2. L’ $\leq_p$ L for every L’ Є NP

NB: If a language L satisfies property 2, but not necessarily property 1, we say that L is NP-hard.

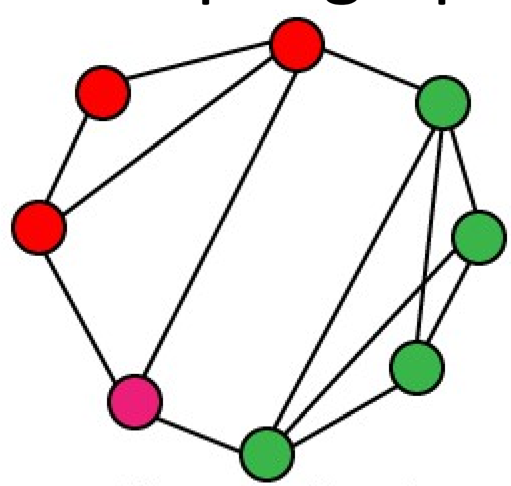
3 CNF is NP-Complete

- A 3-CNF formula φ is **a Boolean formula in conjunctive normal form with exactly three literals per clause**, like

  $\varphi := (x1 \vee x2 \vee \neg x3) \wedge (\neg x2 \vee x3 \vee \neg x4) := \psi 1 \wedge \psi 2$
- A 3-CNF is said to be satisfiable if it has a satisfying assignment and it is called 3 SAT problem.
- **3 CNF is NP-Complete.**

CLIQUE

- Clique in an undirected graph G = (V, E) is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E.
- A clique is a complete subgraph of G.
- Below is a 3-clique graph.



Clique cover (size=3)

CLIQUE is NP-complete

- To show that CLIQUE NP-Complete, prove :
  - a) CLIQUE $\in$ NP
  - b) Clique problem is NP-Hard

- To prove clique problem to be NP-Hard, show that 3-CNF-SAT $\leq_p$ CLIQUE.

CLIQUE Є NP

- A naïve algorithm for determining whether a graph G=(V,E) has a clique of size k is to list all k subsets of V, and check each one to see whether it forms a clique.
- We use the set V '⊆ V of vertices in the clique as a certificate for G.
- We can check whether V' is a clique in polynomial time by checking whether, for each pair u,v Є V ', the edge (u,v) belongs to E.
- Since it is verifiable in polynomial time, it is a NP problem
- So CLIQUE Є NP

To prove Clique problem is NP-Hard

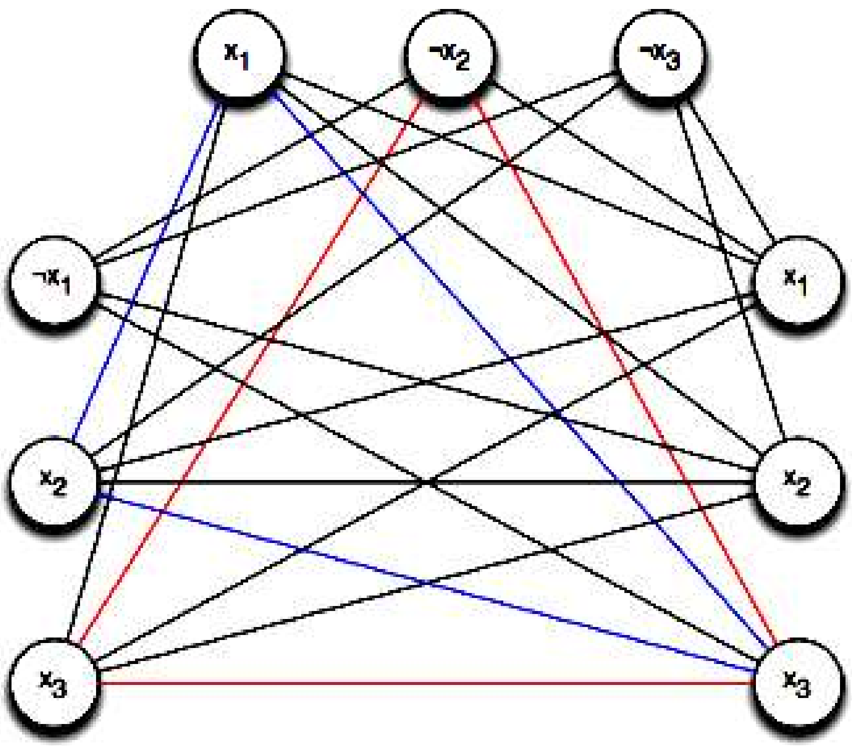
- For that we have to prove that

  3-CNF-SAT $\leq_p$ CLIQUE

- Let Φ= C1 ^ C2 ^ ……^ Ck be a boolean formula in 3-CNF with k clauses.
- For r = 1, 2,….., k, each clause Cr has exactly three distinct literals l1 , l2 , and l3.
- Next we have to construct a graph G such that Φ is satisfiable if and only if G has a clique of size k.

To prove Clique problem is NP-Hard

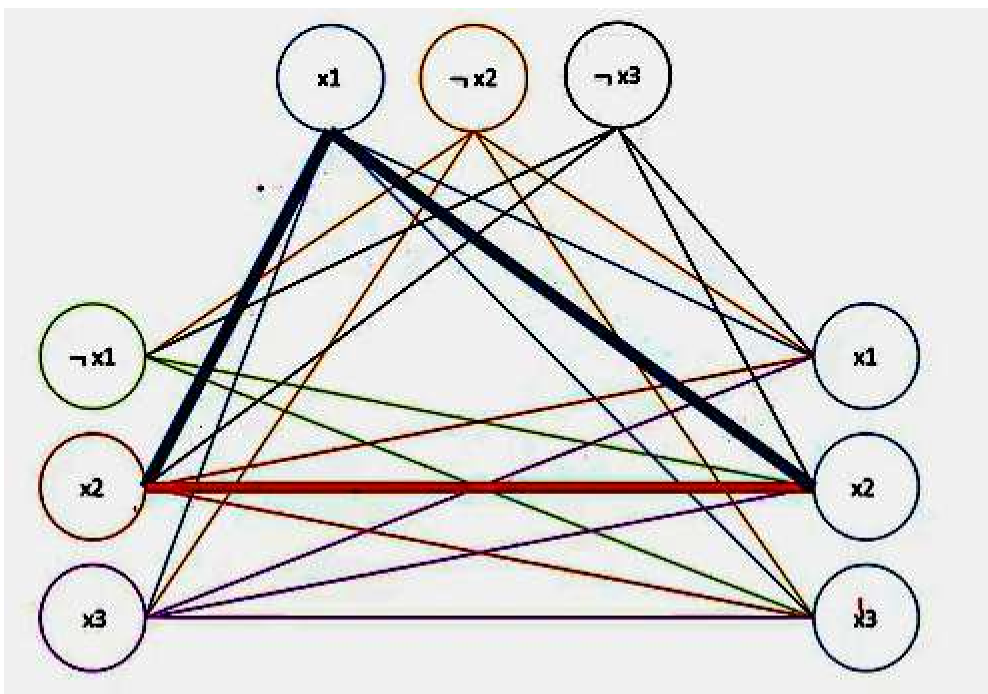
- Consider an example, if we have $\Phi = (x1 \vee \neg x2 \vee \neg x3) \wedge (\neg x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3)$
- This can be converted to a graph G as shown in Figure below.



To prove Clique problem is NP-Hard

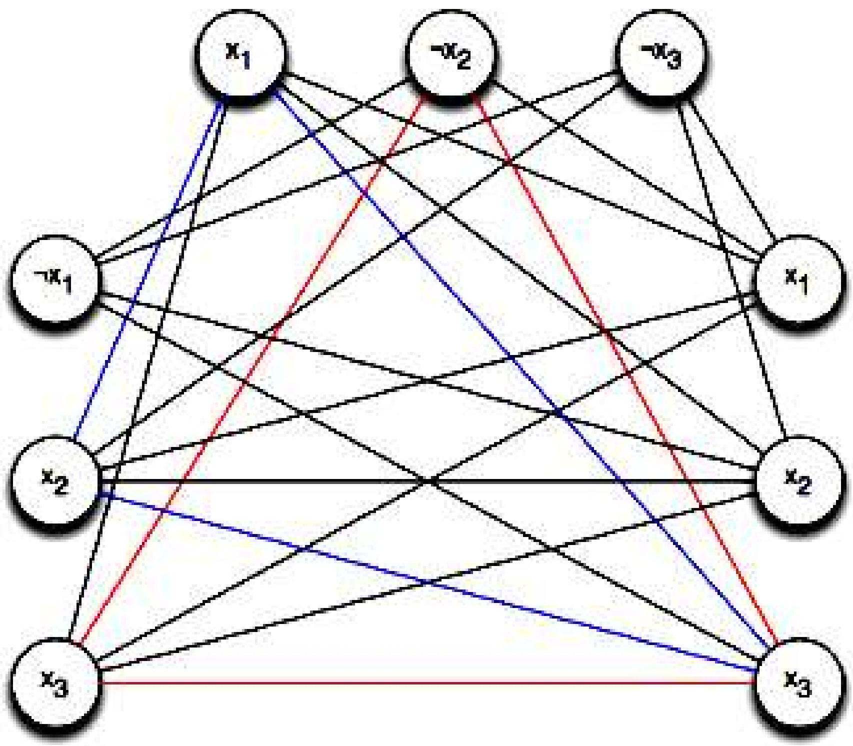
Φ = (x1 v ¬x2 v ¬x3) ^ (¬x1 v x2 v x3) ^ (x1 v x2 v x3)

- From the graph itself we can find cliques.
- Suppose x1=1, x2=1 and x3=0, the above eqn is satisfiable.ie, result of above eqn is 1.



Clique problem is NP-Hard

- Like that we can find different cliques as in below figure.



CLIQUE is NP-complete

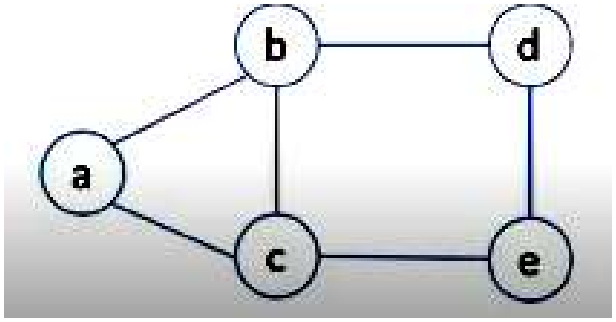
- So we have converted satisfiability problem to clique problem.
- Ie, **3-CNF-SAT $\leq_p$ CLIQUE.**
- **3-CNF-SAT** is a known **NP-Hard problem.**
- So **CLIQUE is also NP-Hard** according to polynomial reduction.
- So **CLIQUE** is **NP & NP-Hard** problem, thus **CLIQUE is NP-complete.**

Vertex Cover Problem

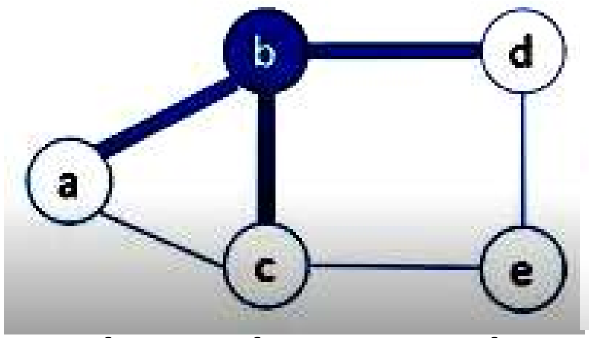
- A vertex cover of an undirected graph G = (V, E) is a subset V '⊆ V such that if (u,v) Є E, then u Є V' or v Є V' (or both).
- ie, each vertex "covers" its incident edges, and a vertex cover for G is a set of vertices that covers all the edges in E.
- The size of a vertex cover is the number of vertices in it.

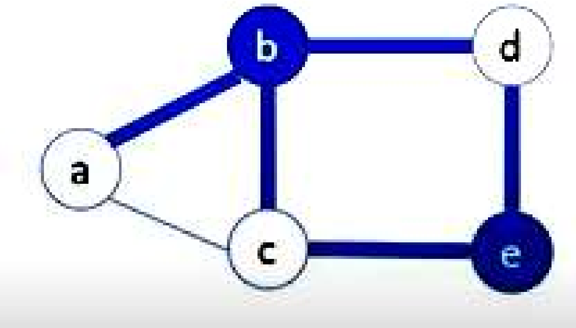
Vertex Cover Problem

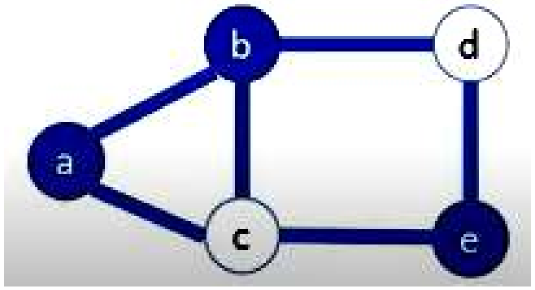
- Consider the graph



- Vertex b covers three edges, e covers 2 edges and a covers 2 edges.







- Thus this graph is having a vertex cover of size 3.
- **Vertex cover problem is to check whether a graph has a vertex cover of size k.**

Vertex Cover Problem is NP-complete

- To show that **Vertex Cover Problem is** NP-Complete, prove :

    a) Vertex Cover Problem $\in$ NP

    b) Vertex Cover Problem is NP-Hard

- To prove Vertex Cover Problem to be NP-Hard, show that CLIQUE $\leq_p$ Vertex Cover.

(CLIQUE is a known NP Complete problem)

Vertex Cover Problem Є NP

- A naive algorithm checks whether the set V' is a vertex cover of size k using the following strategy (for a graph G(V, E)):

```
let count be an integer
set count to 0
for each vertex v in V'
    remove all edges adjacent to v from set E
    increment count by 1
    if count = k and E is empty
    then
        the given solution is correct
    else
        the given solution is wrong
```

Vertex Cover Problem Є NP

- This can be done in polynomial time.
- Since it is verifiable in polynomial time, it is a NP problem.
- So, Vertex Cover Problem Є NP

Vertex Cover Problem is NP-Hard

- This reduction uses the idea of the "complement" of a graph.
- Given an undirected graph G = (V, E), we define the complement of G as G' = (V, E'), where E' = {u,v): u, v ЄV, u ≠ v and (u,v) ∉ E
- In other words, G' is the graph containing exactly those edges that are not in G.
- The output of the reduction algorithm is the instance <G', |V| − k > of the vertex-cover problem.

- To complete the proof, we show that this transformation is indeed a reduction:

  The graph G has a clique of size k if and only if the graph G has a vertex cover of size |V| − k.

Vertex Cover Problem is NP-Hard

- Suppose that G has a clique $V' \subseteq V$ with $|V'| = k$.
- We claim that $V - V'$ is a vertex cover in G.
- Let (u, v) be any edge in E.
- Then, $(u, v) \notin E$, which implies that at least one of u or v does not belong to V', since every pair of vertices in V' is connected by an edge of E.

- Equivalently, at least one of u or v is in $V - V'$, which means that edge (u, v) is covered by $V - V'$.

- Since (u, v) was chosen arbitrarily from E, every edge of E is covered by a vertex in $V - V'$.

- Hence, the set $V - V'$, which has size $|V| - k$, forms a vertex cover for G.

Vertex Cover Problem is NP-Hard

- Conversely, suppose that G has a vertex cover $V' \subseteq V$, where $|V'| = |V| - k$.

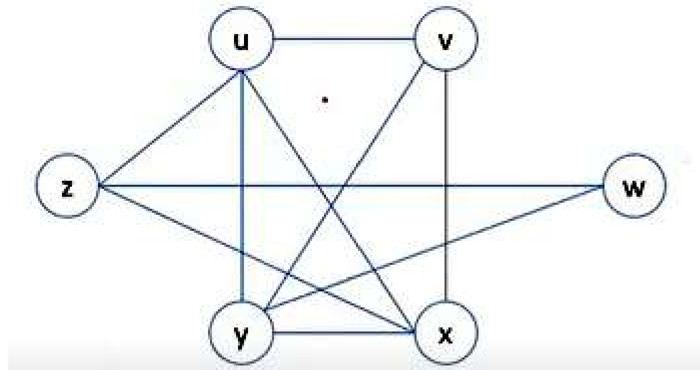
- Then, for all $u, v \in V$, if $(u, v) \in E$, then $u \in V'$ or $v \in V'$ or both.

- The contrapositive of this implication is that for all $u, v \in V$, if $u \notin V'$ and $v \notin V'$,

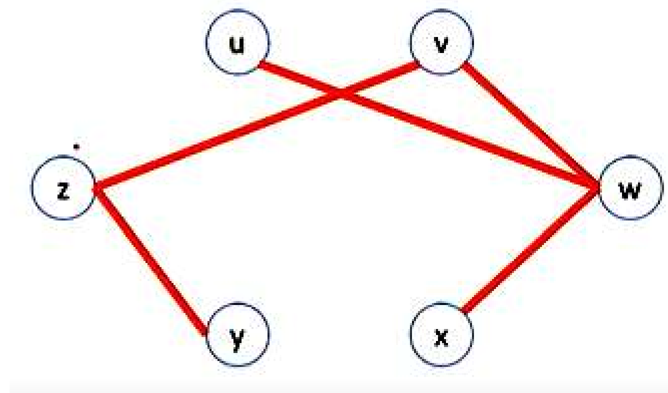
- then $(u, v) \in E$. In other words, $V-V'$ is a clique, and it has size $|V|-|V'| = k$.

Vertex Cover Problem is NP-Hard

- Consider a graph G which has a clique of size 4.ie, {u,v,x,y}



Its complement G’ is ,



- This graph has a vertex cover of 2 ,ie,{ z, w}
- **Thus graph G has a clique of size k if and only if the graph G’ has a vertex cover of size |V| – k.**

Vertex Cover Problem is NP-Hard

- Thus we reduced a clique problem to vertex cover problem.
- So, Vertex Cover Problem is NP-Hard
- Hence Vertex Cover problem is NP-Complete.