

① PCS is polytime reducible to ILP.

PCS

- I/p → i) Directed Acyclic Graph (G) {Vertex → Job (unit time)}
 ii) No. of processors (p)
 iii) Deadline (d)

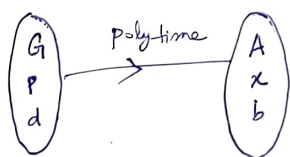
Decision Problem → Can G be scheduled in p processors such that the execution completes in d steps?

ILP

- I/p → i) Matrix A ($m \times n$)
 ii) Vector b , ($m \times 1$)

Decision Problem → Does there exist a vector ' x ' ($n \times 1$) s.t.
 $Ax \leq b$, $x \in \{0, 1\}$

Reduction



In PCS are $(u, v) \rightarrow u$ must execute before v .
 i.e. $T(u) < T(v)$

Consider u & v

t_u = Times at which vertex u is scheduled.

$x_{ut} = 1$, if u is scheduled at time t .
 0, else

Constraint → i) Everytime at step t , only ' p ' nodes (Jobs) can be scheduled.

i.e. $\sum_u x_{ut} \leq p \quad \forall \text{ all } t$. (At time t , max of p nodes can be scheduled as max processors available is p)

$$\Rightarrow x_{1t} + x_{2t} + x_{3t} + \dots + x_{nt} \leq p$$

(assume we have a total of n nodes)

ii) Every Node must be scheduled exactly at one instant. (Every job needs to be executed without any preemption)

$$\sum_t x_{ut} = 1 \longrightarrow O(d)$$

$$\Rightarrow \sum_t x_{ut} \leq 1$$

Now,

Consider the arc (u, v) i.e. u executed before v .

$$\text{Time at which } u \text{ is scheduled} = 1 \cdot x_{u1} + 2 \cdot x_{u2} + \dots + d \cdot x_{ud}$$

\downarrow
 { if u is scheduled at time z ,
 then $z.1 = z$ }
 \downarrow
 [\because deadline is d , u must
 be executed at time $\leq d$]

Since, u must execute before v ,

$$\therefore 1 \cdot x_{u1} + 2 \cdot x_{u2} + \dots + d \cdot x_{ud} < 1 \cdot x_{v1} + 2 \cdot x_{v2} + \dots + d \cdot x_{vd}$$

$$\Rightarrow 1 \cdot x_{u1} + 2 \cdot x_{u2} + \dots + d \cdot x_{ud} \leq 1 \cdot x_{v1} + 2 \cdot x_{v2} + \dots + d \cdot x_{vd} - 1$$

$$\longrightarrow O(1)$$

\downarrow
 (let u execute at time 4 ,
 then v can execute at time 5 ,
 so, $4 \leq (5-1)$)

$$\text{Time Complexity} = O(d + v) = O(n), \text{ Polytime}$$

Thus, PCS can be reduced to the form $Ax \leq b$ (ILP).

$$x = \begin{bmatrix} x_{u1} \\ x_{u2} \\ \vdots \\ x_{ud} \end{bmatrix}, x \in (0, 1).$$

$$\text{PCS} \leq_p \text{ILP}.$$

② Undirected Ham-Cycle is NPC.

Ham-Cycle \rightarrow Cycle in undirected graph $G=(V,E)$ that covers every vertex exactly once excluding the source vertex.

Decision Problem \rightarrow Given a graph $G=(V,E)$, does it contain a Ham-cycle consisting of all the vertices of V ?

To prove NP

x : Sequence of vertices forming Ham-cycle i.e. Set $V' = \{v_1, v_2, \dots, v_n\}$

x : instance of the graph $G=(V,E)$, $|V|=n$

Verifier $V(x,y)$ will check whether all vertices in $V' \subseteq V$ & each pair of vertices in V' are adjacent.

$V(x,y)$: i) if $|V'| \leq |V|$, then return false.

$\because V'$ will contain the source vertex twice

ii) for every ^{adjacent} pair of vertices $\{u,v\}$ in V' :
check if edge $(u,v) \notin E$,

~~if~~ return false

return true.

$V(x,y)$ can verify the solution V' in $O(V+E)$ i.e. polynomial time.

\therefore Ham-Cycle Decision problem is in NP.

To prove NP-C

Reduction \rightarrow (Ham-Path \leq_p Ham-Cycle.)

Every instance of the Ham-path problem in graph $G=(V,E)$ can be reduced to Ham-cycle problem in graph $G'=(V',E')$.

Construction of G' \rightarrow

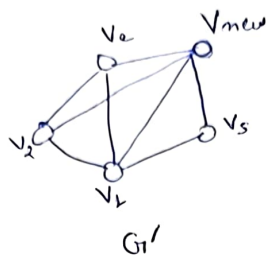
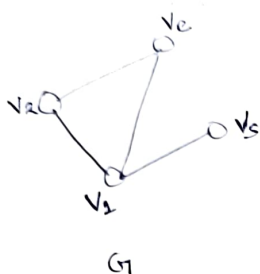
i) Add V vertices of the graph G along with a new vertex

$V_{\text{new}} - (|V'| = |V| + 1)$

ii) Add edges E of the graph G in G' . Now add new edges between V_{new} and remaining $|V|$ vertices in G' such that V_{new} is connected to all the $|V|$ vertices. $(E' = E + |V|)$

Now, assume Graph G has a Ham-path covering (V) vertices by starting at V_s and ends at V_e . Extend the Ham-path to Ham-cycle by the edges (V_e, V_{new}) and (V_{new}, V_s) in G' . Thus G' will now have a Ham-cycle traversing all the vertices once.

G' will have a Ham-cycle iff G has a Ham-path.



$\{V_s, V_1, V_2, V_e\} \rightarrow \text{Ham path in } G$

$\{V_s, V_1, V_2, V_e, V_{new}, V_s\} \rightarrow \text{Ham-cycle in } G'$

This reduction can be done in $O(V+E)$ polynomial time.
i.e. $\text{Ham path} \leq_p \text{Ham-cycle}$.

③ Vertex-Cover is in NPC.

Vertex-Cover \rightarrow subset of vertices that covers all the edges in the graph.

Decision Problem \rightarrow Given an instance of $G=(V,E)$ and the integer ' k ', does there exist a Vertex cover of size k ?

Proof NP

x : Instance of Graph $G=(V,E)$, Integer k

y : Subset of vertices $V' = \{V_1, V_2, \dots, V_k\}$

Verification $V(x,y)$ will check whether the vertices in V' covers all the edges in G .

$V(x,y)$: i) if $|V'| \neq k$,

return false.

ii) for each vertex v in V' :

Remove all edges adjacent to v from E .

if $|E| \neq 0$

return false

return true

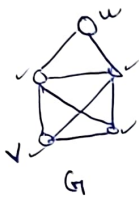
Verifier $V(vy)$ can verify the solution V' in $O(E+V)$ polynomial time.

\therefore Vertex cover $\in NP$.

To prove NP-c, (Clique \rightarrow Vertex-Cover)

Given graph $G=(V,E)$, K does there exist a clique of size K .

To show that any instance (G,K) of clique problem can be reduced to an instance of the vertex-cover problem.



clique = $\{u, v\}$



Vertex-cover = $\{5-3\} = \{2\}$
 $|V|-K$

Consider Graph $G=(V,E)$ of the Clique problem. $\overline{G}^{(V,\overline{E})}$ is the complement of G . Problem of finding whether a clique of size 'K' exist in G is same as problem of finding whether a Vertex-cover of size $|V|-K$ in \overline{G} .

Construction - Assume there exist clique of size K in G .

Set of vertices in clique be V' . i.e. $|V'| = K$

In \overline{G} , consider any edge (u,v) . Then at least one of u or v must be in set $V-V'$. Because if both u & v are in $V' \rightarrow$ edge $(u,v) \in E(G)$ i.e. edge (u,v) is in G which contradicts the fact that $(u,v) \in E(\overline{G})$.

Thus, all edges in \overline{G} are covered by vertices in the set $V-V'$ i.e. size of set = $|V|-K$.

Now, Assume there is a vertex cover V'' of size $|V|-K$ in \overline{G} . So, all edges in \overline{G} are connected to some vertex in V'' . This means for any edge (u,v) of \overline{G} , both u & v cannot be outside of V'' .

All the edges (u,v) s.t. $u,v \notin V''$ are in G and form a clique of size $|V| - (|V|-K) = K$.

So, There exist a clique of size k in G iff there exist a vertex cover of size $|V| - k$ in \overline{G} .

Hence instance of Clique problem can be reduced to instance of Vertex-cover problem.

For generating \overline{G} , we need $O(n)$ time.

So, $\text{Clique} \leq_P \text{Vertex-Cover}$.

④ Set-Cover is in NPC

Set-Cover \rightarrow Consider finite set S . Set cover is the ^{subset of} set of subsets $\{S_1, S_2, \dots, S_n\}$ s.t. $S_1 \cup S_2 \cup \dots \cup S_n = S$.

Decision Problem \rightarrow Given a finite set S , ^{collection of} subsets S_1, S_2, \dots, S_n and +ve integer k , does there exist a set-cover of size k ? (i.e. $S_1 \cup S_2 \cup \dots \cup S_k = S$)

Proof NP

$x: S, \{S_1, S_2, \dots, S_n\}, k$

$y: \text{Set of subsets } S' = \{S'_1, S'_2, \dots\}$

$V(x, y):$ i) if $|S'| \neq k$
return false.

ii) for all the elements in $S'_i, \forall i = 1 \text{ to } k$
mark elements in S \forall element $\in S'_i \rightarrow O(n)$

iii) for all elements in S ,
if any element is not marked $\rightarrow O(n)$
return false
return true.

$V(x, y)$ can perform the verification in $O(n)$ time.

$\therefore \text{Set-Cover} \in \text{NP}$.

Proof NPC

(Vertex-Cover \rightarrow Set-Cover)

Consider the problem instance of Vertex-Cover of size k in $G \in \text{NP}$.

Construct \rightarrow an instance of Set-Cover \rightarrow

i) $S = E$ (Set of all edges in G be S)

ii) $S_i = \{(u,v) \in E \mid u=i \text{ or } v=i\} \forall i \in V$.

(Create subset S_i for every vertex i consisting of all the edges incident to i)

iii) $K = K$ (Size of Set cover = size of Vertex cover)

\rightarrow The vertices corresponding to the subsets in Set cover of S will be vertex cover of G .

Validation \rightarrow

Assume V' of size K is the ^{Vertex cover} ~~solution~~ in Graph $G=(V,E)$.

So, vertices in V' cover all the edges in E .

\therefore Any subset S_i is the set of edges incident on vertex i ,

Union of all S_i s.t. $i \in V'$ must contain all the edges in E .

\therefore all the edges in E form the set S , $\{S_i, i \in V'\}$ is the set cover of S , with size K .

Conversely,

Consider solution to the Set cover problem $(S, \{S_1, \dots, S_n\}, K)$.

Solution be the set of subsets, $S' = \{S_1, S_2, \dots, S_K\}$,

Then $S_1 \cup S_2 \cup \dots \cup S_K = S$.

$\therefore E = S$, S' covers all the edges in E .

\therefore each $S_i \in S'$ are the edges incident on vertex i ,

the vertices $\{i : S_i \in S'\}$ forms a vertex cover of G with size K .

This reduction can be done in polynomial time (need to scan all the edges of G to create subsets S_i & S).

$O(E)$.

\therefore Vertex Cover \leq_p Set-Cover.

⑤ Clique is in NPC

Clique \rightarrow Subgraph of a graph G in which all the vertices are connected. (Complete subgraph).

Decision Problem \rightarrow Given a graph $G=(V,E)$ and integer K , does G contain a clique of size K ?

Proof NP

x : instance of graph $G=(V,E)$, K .

y : set $V' \subseteq V(G)$.

$V(x,y)$: i) if $|V'| \neq K$,
return false.

ii) for every pair of vertices (u,v) , $u \neq v$, $u,v \in V'$
if $\text{edge}(u,v) \notin E$ or $\text{edge}(u,v)$ do not exist
return false.

return true.

This verification can be done in polynomial time.

Proof NPC (3-SAT \rightarrow Clique)

Consider the instance of 3-CNF-SAT.

Let $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_k$ be a boolean formula in 3-CNF with k clauses.

Each clause C_π , $\pi=1,2,\dots,k$ has 3 distinct literals $l_1^\pi, l_2^\pi, l_3^\pi$.

Graph $G=(V,E)$ is constructed as below \rightarrow

1) For each clause $C_\pi = (l_1^\pi \vee l_2^\pi \vee l_3^\pi)$ in ϕ , create 3 vertices in V corresponding to each literal in C_π .

2) Create edge between the vertices x when —
 V_i^π & V_j^s

a) V_i^π and V_j^s are in different clauses, i.e. $\pi \neq s$. (—)

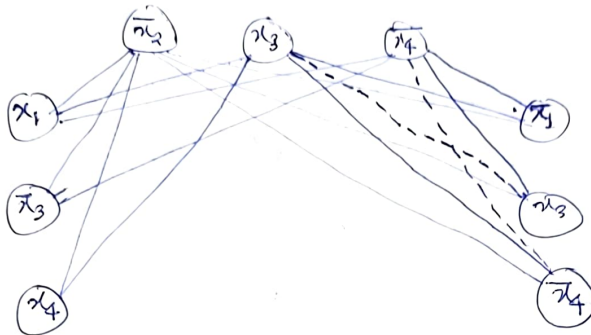
b) their corresponding literals are consistent, i.e. (---)
 $l_i^\pi \neq \bar{l}_j^s$ (or when $l_i^\pi = l_j^s$ or $\bar{l}_i^\pi = \bar{l}_j^s$)

This graph can be computed in polynomial time.

$G_1 = (V, E)$ is 3-SAT if ϕ is satisfiable iff G_1 has clique of size k .

- constr.)
- i) for each literal, create a vertex.
 - ii) Connect each literal x_i or \bar{x}_i to every other literal x_j or \bar{x}_j where $i \neq j$ in every other clause.
 - iii) Connect each $x_i \rightarrow x_i$ of other clause, $\bar{x}_i \rightarrow \bar{x}_i$ of other clause.

eg: $(x_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$



Validation \rightarrow Suppose ϕ has a satisfying assignment.

Each clause C_k contains atleast 1 literal L^k that is assigned 1.

Such literal corresponds to vertex V_i^k .

From k clauses, k vertices^{set} can be formed s.t $|V'| = k$.

Claim $\rightarrow V'$ is a clique.

- For any 2 vertices $V_i^k, V_j^s, i \neq j$, corresponding literals L^k & L^s are mapped to 1 by satisfying assignment and thus literals cannot be complements. So edge

$$(V_i^k, V_j^s) \in E(G).$$

Conversely,

Suppose G has clique V' of size k .

V' contains exactly one vertex per clause as no edges in G connects vertices of same clause.

Assign 1 to the corresponding literal L^k if $v \in V'$ without worry of assigning 1 to both literal & its complement as no edges between inconsistent literals.

Each clause is satisfied and hence ϕ is satisfied.

$$\therefore 3\text{-CNF} \leq_P \text{Clique}.$$

② TSP is NPC

TSP → A salesman has to visit 'n' cities visiting each city exactly once. Cost is associated for travelling from city i to j. The salesman wants to complete the tour with minimum cost.

Decision Problem → Given graph $G_n = (V, E)$, Cost C is function from $V \times V \rightarrow \mathbb{Z}$, $K \in \mathbb{Z}$ does G has a Travelling Salesman tour with cost at most K?

NP Proof

$V(x, y):$

$x \rightarrow$ instance of ^{weighted} $G = (V, E)$, K

$y \rightarrow$ Sequence of 'n' vertices, $V' = \{v_1, v_2, \dots, v_n\}$

$V(x, y):$ i) For each vertex in V' ,

check if any vertex repeats
return false

ii) For each pair of adjacent vertices $u_i, u_j \in V'$, $i \neq j$ till

$sum = sum + C(u_i, u_j)$
(initial 0)

if $sum \leq K$
return false

return true.

$V(x, y)$ can verify in polynomial time.

NPC Proof (Ham Cycle \rightarrow TSP)

Consider instance of the HAM Cycle problem with graph $G = (V, E)$

Construct an instance of the TSP as below \rightarrow

i) Form Complete graph $G' = (V, E')$, $E' = \{(i, j) : i, j \in V, i \neq j\}$

ii) Define Cost function $C(i, j) = \begin{cases} 0, & (i, j) \in E \\ 1, & (i, j) \notin E \end{cases}$

This construction can be done in polynomial time.

Claim \rightarrow Graph G has a Ham cycle iff graph G' has a tour of cost at most 0.

Validation \rightarrow

Suppose G has Ham-cycle h . Each edge in ' h ' belongs to E and has cost 0 in G' . Thus ' h ' is a tour in G' with cost 0.

Conversely,

Suppose graph G' has a tour ' h ' of cost at most 0.

Since the cost of edges in E' are 0 or 1, the cost of tour ' h ' is exactly 0 and each edge on the tour must have a cost 0. So, ' h ' contains only edges in E .

Thus ' h ' is a ham cycle in graph G .

\therefore Ham Cycle \leq_p TSP.

② Longest Simple Cycle is NPC

Problem \rightarrow Determining a simple cycle (no repeated vertices) of maximum length in a graph.

Decision Problem \rightarrow Given graph $G=(V,E)$, integer K does there exist a simple cycle of length at least K ?

NP Proof

x : Instance of graph $G=(V,E)$, K

y : Sequence of ' K ' vertices, $V' = \{v_1, v_2, \dots, v_K\}$

$V(x,y)$: i) For each vertex in V' ,
check if any vertex repeats
return false if YES.

ii) For each pair of adjacent vertices $v_i, v_j \in V'$, $\forall j=i+1$,
check if edge $(v_i, v_j) \notin E$ till $j=K$
return false

return true

$V(x,y)$ takes linear time $O(\text{no. of vertices})$ i.e. polynomial time.

NP proof. (Ham Cycle \rightarrow Longest simple cycle)

Consider the instance of Ham Cycle problem with graph $G=(V,E)$ and $|V|=K$.

Construct a new graph G' containing the same graph G , i.e. $G'=(V,E)$ with $K=|V|$.

Claim \rightarrow If Graph $G=(V,E)$ has a Ham cycle on $|V|$ vertices iff G' has a longest simple cycle of length K .

Assume G has a Ham cycle on $|V|$ vertices. So the cycle covers all the vertices of graph G exactly once. Consequently, graph G' will have a longest simple cycle of length K .

Conversely,

Assume G' has a longest simple cycle of length K . So the cycle covers all the $K=|V|$ vertices exactly once and thus G will have a Ham Cycle on $|V|$ vertices.

Graph G' construction will take polynomial time.

⑧ 3-Coloring is NP

K-coloring problem \rightarrow Assignment of colors to the nodes of a graph s.t. no two adjacent vertices have same color and atmost K colors can be used.

Decision Problem \rightarrow Given Graph $G=(V,E)$ and $K=3$, can the graph be colored using atmost 3 colors s.t. no two adjacent vertices are assigned same color.

Proof NP

x : instance of graph $G=(V,E)$.

y : E is the list of colors ~~in some order~~ with vertices.

Assignment of colours $\{c_1, c_2, c_3\}$ where each vertex $v \in V$ is assigned one colour.

$N(x,y)$: i) For each edge (u,v) in G , $u,v \in V$ & $u \neq v$
check if $\text{Color}(u) \neq \text{Color}(v)$

return false

return true.

$\rightarrow O(E)$ Polynomial.

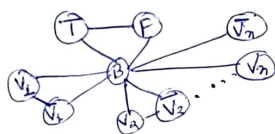
NP Proof ($3\text{-SAT} \rightarrow 3\text{-Color}$)

Consider the 3-SAT problem having 'm' clauses on 'n' variables.
Let the variables be x_1, x_2, \dots, x_n .

Construct a graph from the formula as follow-

- i) for every variable x_i , create vertex v_i and a vertex v_i' denoting \bar{x}_i .
- ii) for each clause c in m , add 3 vertices j_1, j_2, j_3 .
- iii) Create 3 special vertices denoting True, False and Base (T, F, B) .
- iv) Edges are added among T, F, B to form a triangle.
- v) Edges are added among v_i and v_i' and B to form a Δ .

Constraints -

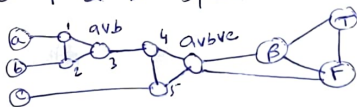


- i) For each pair of vertices v_i & v_i' , exactly one is assigned TRUE and other FALSE.
- ii) For each clause c in 'm' clauses, atleast one of the literals in the clause is assigned TRUE.

So, A OR-gadget graph can be constructed for each clause $C = (u \vee v \vee w)$ in the formula by i/p nodes u, v, w and connect output node of the gadget to both F & B special nodes.

Validation

for clause $C = a \vee b \vee c$



Assume 3-SAT formula has satisfying assignment, then in every clause atleast one of the literals x_i has to be TRUE.

Then corresponding v_i can be assigned TRUE & \bar{v}_i FALSE.

Extending this, for each clause the corresponding OR-gadget graph can be 3-coloured. Hence, graphs can be 3-coloured.

Conversely,

if graph is 3-colourable, so if the vertex v_i is assigned TRUE, corresponding variable x_i is assigned TRUE. Hence, there is

a satisfying assignment to 3-SAT clause.

So, $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m = \text{TRUE}$

⑨ 0-1 ILP is NPC

D. Problem → Given an integer Matrix $A (m \times n)$ and an integer Vector $b (m \times 1)$, does there exist an integer Vector $x (n \times 1)$ with elements $\{0, 1\}$ s.t $Ax \leq b$?

Proof NP

x : Matrix $A (m \times n)$, Vector $b (m \times 1)$

y : Vector $x (n \times 1)$

$V(x, y)$ — i) Compute Ax — $O(mn)$

ii) Compare elements of Ax and b — $O(m)$

if any element in $Ax > b$

return false

return true.

Verifier can verify the solution in $O(mn)$ i.e polynomial time.

Proof NPC ($3\text{-CNF-SAT} \rightarrow 0\text{-1 ILP}$)

Consider the problem instance of 3-CNF having formula ϕ containing ' v ' variables and ' c ' clauses.

Construct $(c \times v)$ matrix A s.t

$$a_{ij} = \begin{cases} -1, & \text{if variable } j \text{ occurs ^{only} without negation in clause } i \\ 1, & \text{" " " ^{only} with negation " " } i \\ 0, & \text{otherwise} \end{cases}$$

Construct vector $b (c \times 1)$ s.t

$$b_i = -1 + \sum_{j=1}^v \max \left(0, a_{ij} \right) \quad \text{no. of negated literals in clause } i$$

Construction of A & $b \rightarrow O(cL)$ polynomial time.

Claim — There exist vector $x (v \times 1)$ s.t $Ax \leq b \Leftrightarrow \phi$ is satisfiable.

Consider x to represent an assignment of variables in ϕ ,

$$x_i = \begin{cases} 1, & \text{if variable } i \text{ is assigned TRUE} \\ 0, & \text{if variable } i \text{ is assigned FALSE} \end{cases}$$

Let $y = Ax$, then

$y_i \leq b_i \Leftrightarrow$ sum of satisfied literals in clause $i \geq 1 \Leftrightarrow$
clause i is satisfied.

$\therefore y = Ax \leq b \Leftrightarrow x$ is an assignment satisfying ϕ .

⑩ Independent Set is NPC

Inst \rightarrow Set S of Graph $G=(V,E)$ is a set of vertices s.t.
no 2 vertices in S are adjacent to each other.
It consist of non-adjacent vertices.

Factorem \rightarrow Given a graph $G(V,E)$ and integer k , does there
exist an independent set of size $\geq k$?

No Proof

x : instance of graph $G=(V,E)$, k

y : Set of vertices $S = \{v_1, v_2, \dots, v_k\}$

$V(x,y)$: i) if $|S| < k$
return false.

ii) for every edge in G , check

$(u,v), u \neq v$

if $u, v \in S$ (both vertices are in S)

return false

return true.

Proof NPC (Clique \rightarrow IS)

Consider the clique problem consisting of graph $G(V,E)$ and
integer k .

Construct graph G' as below $\rightarrow O(V+E)$
 (V', E')

$V' = V$

$E' =$ complement of edges E , i.e. edges not present in G .

G' is complement of graph G .

Graph G has a clique of size k iff G' has IS of size k .

Validation →

Assume, G has clique of size k . This implies there are k vertices in G where each vertex is connected by an edge with the remaining vertices $(k-1)$. Since these edges are in G , \therefore they are not present in G' .
So, these k vertices are not adjacent to each other in G' and hence form an I.S of size k .

Conversely,

Suppose G' has I.S of size k . None of these vertices share an edge with any other vertices. Take complement of G' to obtain G where these k vertices will share edge with each other and hence form a clique of size k in G .