- Vertex Cover = minimum subset of vertices of G which contain atleast one of two endpoints of each edges in G.

Q28 Vertex Cover $\equiv_p$ Independent set. Prove.

ans Part-1 :-

- claim :- Vertex cover $\equiv_p$ Independent set.
- Proof :- We show S is an independent set iff $V-S'$ is vertex cover.

Part - 1 :-

- to prove that, if S' is independent set than $V-S'$ is vertex cover. ($VC \leq_p IS$)

- Proof :-

- Consider an edge (u,v) in E [E is edge set]
- Because S' is an IS, either $u \notin S'$ or $v \notin S'$ third condition was $u \& v \notin S$, but this is not possible as (u,v) is edge present as per our consideration.
- This implies $u \in (V-S')$ or $v \in (V-S')$
- " " , at least u or $v \in (V-S')$ for an edge (u,v) in E.
- Arguing same for all the edges in the graph, $(V-S')$ is vertex cover of graph. Hence proved.

Part 2 :-

- to prove that if $(V-S)$ is vertex cover than S is independent set ($IS \leq_p VC$)

• Proof :-

- Consider two verticies u and v in V.
- What can we argue about edge (u,v) ?

(a) either $u \in (V - S')$ or $v \in (V - S')$

implies $u \notin S'$ or $v \notin S'$

(b) ~~either~~ both $u \in (V - S')$ and $v \in (V - S')$

implies both $u \notin S'$ and $v \notin S'$

(d) ↳ It is possible for valid f

doesn't harm IS property.

(c) both $u \notin (V - S')$ and $v \notin (V - S')$

↳ It is not possible for valid edge, bcoz it will vilocate vertex cover property, eg for valid edge $(u,v) \in E$, if $u \notin VS$ & $v \notin VS$ than ~~vertex~~ who will cover that edge (u,v) ?

- So that for all three possibilities mentioned above, property of IS is maintained. Hence proved.

∴ Hence proved, $VC \equiv_p IS$

Date
Page

Q Vertex Cover $\equiv_p$ k-clique

claim :- Vertex Cover $\equiv_p$ k-clique

Proof :- We show graph $G(V,E)$ has a clique (S) of size k iff complement graph $G_c(V,E')$ has vertex cover of size $|V|-k$.
(V - S)

Part-1 :-

• to prove that, if graph G has a clique (S) of size k, then complement graph G_c has vertex cover $(V-S)$ of size $(|V|-k)$. [VC $\leq_p$ k-clique]

• Proof :-

for an edge (u,v) in G_c, there are following cases in ~~graph~~ G.

case 1 :- u & $v \in S$.
Is it possible ? No
because (u,v) is edge in G^c. So there ~~will~~ would not be an edge (u,v) in G. and as there is not an edge betⁿ u and v, both of them can not ~~be formed~~ be part of clique in G together.

case 2 :- either $u \in S$ or $v \in S$
Is it possible ? Yes
implies either $u \notin (V-S)$ or $v \notin (V-S)$
" " $v \in (V-S)$ or $u \in (V-S)$
therefor $(V-S)$ is vertex cover in G_c ~~for~~ w.r.t an edge (u,v) in G_c

case 3 :- u & $v \notin S$, Possible ? Yes → u & $v \in (V-S)$ therefore $(V-S)$ is VC in G_c w.r.t an edge (u,v) in G_c.

Date
Page

$\therefore$ We argue above for all the edges in G_c and Hence the proof that G_c has vertex cover $(V-S)$ of size $(|V|-k)$.

~~Part 2~~ :-

~~to prove that if~~

And for edge (u,v) not in G_c, we don't thinking about that because it doesn't obey definition of vertex cover.

Part - 2 :-

- to prove that, if graph G_c has vertex cover $(V-S)$ of size $(|V|-k)$, than complement graph G has clique (S) of size k. [k-clique & VC]

- Proof :-

Consider an edge (u,v) in G_c.

case 1 :- if $(u,v) \in G_c$ then either $u \in (V-S)$ or $v \in (V-S)$

then by contrapositive,

if $u \notin (V-S)$ and $v \notin (V-S)$ then $(u,v) \notin G_c$

$\Rightarrow$ if $u \in S$ and $v \in S$ then $(u,v) \in G$.

therefor S is clique ~~edge~~

Case 2 :- if $(u,v) \in G_c$ then $u \in (V-S)$ & $v \in (V-S)$

$\therefore$ if u & $v \in (V-S)$ then $(u,v) \notin G_c$

$\therefore$ if u & $v \in S$ then $(u,v) \in G$.

therefor S is forming clique.

∴ We argue this for all the edges &
~~Hence the proof~~ that G has Clique of
size k.

∴ Hence the proof that

$$VC \equiv_p k\text{-clique}.$$

Q) Prove that the problem k-clique is NP-Complete.

ans Proof :- In order to prove that k-clique is an NPC, we need to ~~prove~~ prove the following :

1) the problem k-clique belongs to class NP. This can be done using ~~certifier~~ certificate approach.

2) We need to select another problem (SAT in this case) to be belonging to class NP & prove that this "another problem" is polynomially reduces to k-clique problem.

↳ Note that since underlying Datastructure of k-clique (graph G(V,E)) & that of SAT problem are different, we have to design gadget on which our subsequent proof can be based. ~~Hence~~ ~~we~~ ~~need~~ ~~to~~ ~~design~~

~~Hence we first show the design~~ of

(*) Proof that problem k-clique belongs to class NP using certifier-certificate approach.

↳ Certifier - Certificate Approach :- If a problem belongs to class NP, then it should have polynomial-time verifiability, that is given a certificate, we should be able to verify in polynomial time if it is sol^n of the problem.

↳ For k-clique problem,

Certificate :-
Using choise (), we selects k nodes in set S, (S is subgraph of G)

Verification :-
We have to check if there exists a clique of size k in graph. Hence verifying if no. of nodes in S equals k, takes O(1) time. Verifying whether each vertex has an out degree (k-1) takes $O(k^2)$ time. Therefore, to check if the graph formed by k nodes in S is complete or not takes $O(k^2) \in O(n^2)$ ($\because k <= n$).

$\therefore$ Therefore, k-clique decision problem has polynomial time verifiability & hence belongs to class NP.

Date ______
Page ______

(*) Now, lets us design gadget as follows:-

↳ Assume that we are given SAT expression F, consisting of k-clauses. (k=3 in our example as below)

$$F = (x_1 + \overline{x}_2)(x_3 + x_4 + \overline{x}_2)(\overline{x}_1 + \overline{x}_4)$$

↳ Our ~~graph~~ gadget must show the graph modelled as expression F.

↳ Gadget Design Rules :-

(1) Consider all the literals of SAT expⁿ as grouped in classes depending on which class they are in, e.g. we have 3-clauses in given SAT expⁿ.

(2) The no. of vertices in graph is equal to no. of literals.

(3) Label vertices as (σ, i), (δ, j) where σ, δ represent literal & i, j represent clause number.

(4) (a) Do not connect two vertices if they are complement of each other.
(b) Do not connect two vertices ~~they~~ from ~~belon~~ same class.
(c) Connect all other vertices.

i.e, Do not connect (σ, i) & (δ, j) if $\sigma = \overline{\delta}$ or $i = j$.

(*) Proof that SAT ~~reduces~~ $\le_p$ k-clique.

** CLAIM #1 If given SAT exp^n F with k-clauses is satisfiable than corresponding graph is k-clique.

Proof #1 Let us assume that we are given a SAT exp^n F with k-clauses which is satisfiable.

- Let us now pick up one literal each from a clause say $\alpha_1, \alpha_2, \ldots, \alpha_k$ i.e. k literals that are all assigned value 1.

- Let us identify corresponding vertices in graph, noting that vertices must come from $diff^n$ clauses (since each literal is from diff clause) ... as $(\alpha_1, 1), (\alpha_2, 2), \ldots, (\alpha_k, k)$.

- The question is are all these vertices connected?
  We say they are, since as per our design rule, only vertices that are complement of each other or ~~are~~ vertices that are in same class are not connected; remaining all are connected.

- And since we have k-vertices, they are forming k-clique.

Date ______
Page ______

CLAIM #2 :- If a given graph $G(V,E)$ has a clique of size k, then corresponding SAT exp^n is satisfiable.

Proof #2 :- We note that, we are given a graph $G(V,E)$ with k-clique.

- Let us identify vertices of given graph G as $(\alpha_1, 1), (\alpha_2, 2), \ldots, (\alpha_k, k)$. note that the vertices selected are in different clauses because only then can be connected forming k-clique.

- The question is that, can the corresponding literals (to vertices) be assigned truth value '1'?
We say yes because as per our design rule, if vertices are connected they can't be complement of each other.

- And therefore we have SAT expression F with k clauses which is satisfiable.

$\therefore$ Hence k-clique problem is NPC.

Date ______
Page ______

Q) Prove that the problem independent set is NP-C.

ans) Proof: In order to prove that Independent set is NPC, we need to prove following.

1. The problem independent set belongs to class NP. This can be proved using certifier - certificate approach.

2. We need to select another problem (3-SAT in this case) to be belonging to class NP & prove that this "another problem" is polynomial reduces to independent set ~~problem~~ problem.

↳ Note that since underlying data structure of Independent set (graph) & that of 3-SAT problem are diffⁿ, we have to design gadget on which our subsequent proof can be based.

(*) Proof that Independent set belongs to class - NP using certifier - certificate approach.

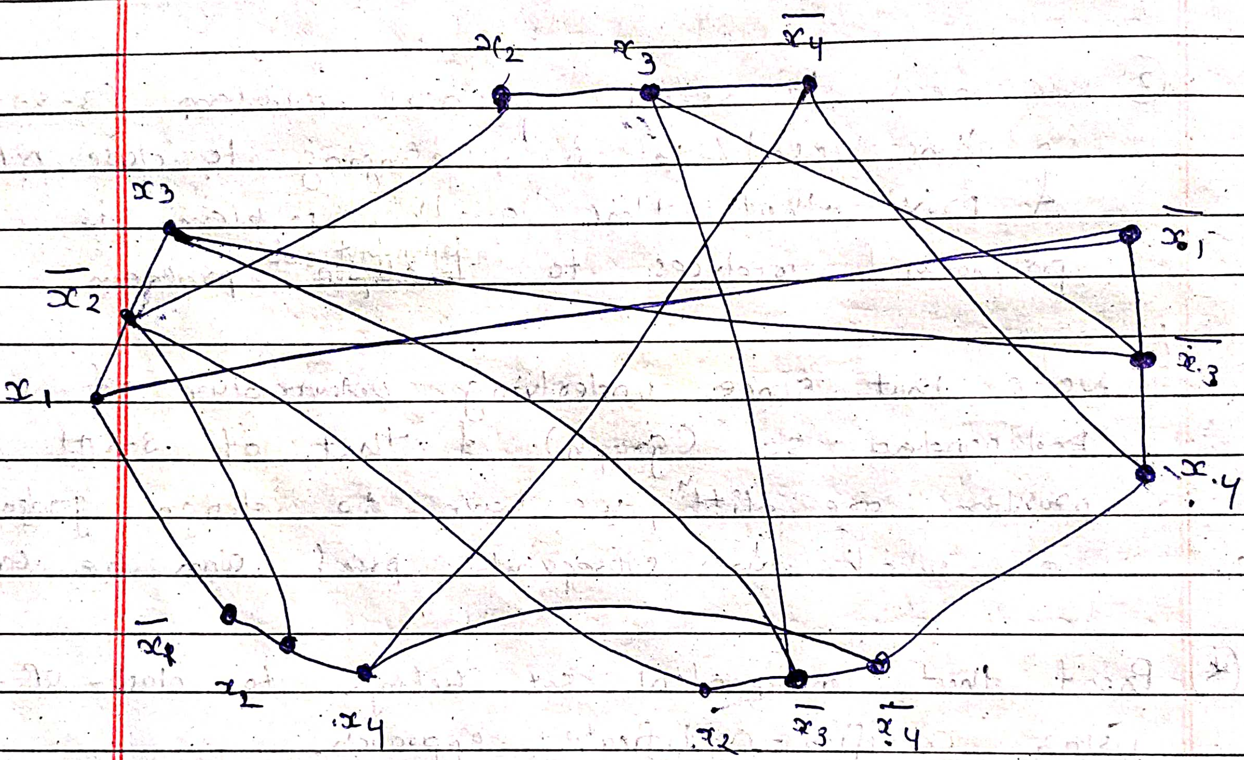
↳ The certificate is a subset V' of vertices, which comprises the vertices belonging to independent set.

(Certifier)
↳ Verification :- We can verify this solⁿ by checking that each pair of vertices belonging to the solⁿ set (V') are non-adjacent, by simply verifying that they don't share an edge with each other. This can be done in polynomial time, that is O(V+E).

* Now, lets us design gadget as follows :

↳ Assume, that we are given 3-SAT expression F, consisting of k clauses, as follows,

$$F = (x_1 + \overline{x_2} + x_3)(x_2 + x_3 + \overline{x_4}).(\overline{x_1} + x_2 + x_4).(x_2 + \overline{x_3} + \overline{x_4})(\overline{x_1} + \overline{x_3} + x_4)$$



↳ Design Rule,

- In graph corresponding to the expression F, the no. vertices is equal to the no. of literals in given 3-SAT exp^n F, i.e $3*|F|$
- However, the vertices pertaining to a clauses are to be drawn as a flattened triangle
- Now, connect two vertices only if they are complement of each other.

Date
Page

(*) Proof that 3-SAT $\leq_p$ k-clique. Independent Set.

** Claim #1 Given a 3-SAT expⁿ F, consisting of m-clauses, the corresponding graph G has an Independent set of size m.

Proof #1 :- Assume that we are given a 3-SAT expⁿ F, consisting of m-clauses which is satisfiable.

- Since F, is satisfiable, we have at least one literal from each clause that is true.

- Lets us identify all such true literals as $x_1, x_2, x_3, \ldots, x_m$. Note that we have such m literals

- Lets us identify the corresponding vertices in gadget. Since each literal comes from diffⁿ ~~for~~ clause, each corresponding vertices will belongs to each distinct flattened triangle.

- Now the question is, "Are these vertices are connected?"

  → We argue, these are not since each corresponding literal is true. & as per our design rule, the vertices are connected only if the corresponding literals are complement of each other.

  → And since these are m such vertices, we have an independent set of size m.

Date
Page

Claim #2 :- Given a graph with an IS of size m, then corresponding 3-SAT exp^n with m clauses is satisfiable.

Proof #2 :- We note that we are given independent set V' of size $\geq m$ of graph G.

:- We can't select two-vertices from a single flattened triangle, and since there are m flattened Δ, V' has exactly one vertex from each flattened triangle. Note that if a vertex labeled x is in V', then adjacent vertex $\bar{x}$ can't also be in V'. Therefore there exists an assignment in which every literal corresponding to a vertex appearing in V' is set to true. Such an assignment satisfies one literal in each clause, and therefore the entire formula is satisfied.

$\therefore$ Hence the proof that Independent Set problem is NP-Complete.

Date ________
Page ________

① Prove that 3 SAT α_p SAT.

ANY

Proof :– let us assume we are given a SAT expression $C = \{c_1, c_2 \ldots c_k\}$ of k-clauses, each of which has literals drawn from a set of boolean variables $U = \{x_1, x_2, \ldots, x_m\}$

↳ We wish to convert C into an equivalent expression C', that may consist of k-clauses, plus some extra additional clauses, if required. But with the restriction that each clause shall have three literals. These literals may be drawn from the same set U plus some additional literals. During conversion it must be insured that, if C is satisfiable C' is also satisfiable and if C is not satisfiable C' is also not satisfiable.

↳ So now let us assume some C_i is of following form

$$C_i = \{z_1\ z_2\ z_3 \ldots z_k\}$$

↳ If $k = 1$, $C_i = \{z_1\}$ & we use additional literals $y_{i,1}$, $\overline{y_{i,1}}$, $y_{i,2}$, $\overline{y_{i,2}}$ in the following form to obtain C_i'

$$C_i' = \{z_1 \vee \overline{y_{i,1}} \vee \overline{y_{i,2}}\} \wedge \{z_1 \vee \overline{y_{i,1}} \vee y_{i,2})$$

$$\wedge (z_1 \vee y_{i,1} \vee \overline{y_{i,2}}) \wedge (z_1 \vee y_{i,1} \vee y_{i,2})$$

Date
Page

↳ if $k = 2$, $C_i = \{z_1\ z_2\}$ and we use $y_{i,1}$, $\overline{y_{i,1}}$ as additional literals in following form to obtain C_i'

$$C_i' = (z_1 \vee z_2 \vee y_{i,1}) \wedge (z_1 \vee z_2 \vee \overline{y_{i,1}})$$

↳ if $k = 3$, $C_i = \{z_1\ z_2\ z_3\}$, we don't need to use any additional literals,

$$C_i' = (z_1 \vee z_2 \vee z_3)$$

↳ if $k \geq 4$,

if z_1 or z_2 is true, assign all additional variables the truth value false.

if z_{k-1} or z_k is true, assign all additional variables the truth value true.

Otherwise, if any middle z_l is true, then assign $y_{i,j}$ the value true for all $1 \leq j \leq l-2$ and assign $y_{i,j}$ the value false for all $l-1 \leq j \leq k-3$.