H9 23

To prove ACX is NP Hard, we need to ~~pro~~ reduce a known NP Hard problem to ACX.

AC is NP complete (as given in question) which makes it NP Hard as well. Now AC can be reduced to Independant set (IS) problem as both are NP complete problems (also IS is covered in class). Now if we can reduce IS, a known NP Hard problem (NP completeness ensures NP Hard) then by chain rule, $AC \propto IS \propto ACX$, ACX will be proved NP Hard from AC. $\Rightarrow AC \propto ACX$.

reduction algorithm

input instance are of IS problem

def reduce (graph G, k)

G' = new empty graph

for all vertices v in G : — O(v)

copy v in G'

for all edges E in G : — O(E)

copy E in G'

for every vertex v' in G' : — O(v') ≈ O(v)

add a new vertex v_{new}

and connect it it v' with an edge

k' = k + |V|

return (G', k')

explanation

we are copying the entire graph G to G'. Then for each vertex v' in G' we are adding a new vertex V_{new} and connecting it with v'. h' of A(x) is the k ~~&~~ integer of IS ~~or~~ added to the number of vertices of G.

running time

running time of reduce algorithm is :—

$$O(v) + O(E) + O(v')$$

$$\Rightarrow \quad O(v) + O(E) + O(v)$$

$$\Rightarrow \quad O(v) + O(E)$$

as shown above, which is in polynomial time.

Lemma

we will get yes instance of ~~AC~~ IS iff we get yes instance of ACX.

If ~~AC~~ IS returns a yes instance ie G contains a set V of atleast n k vertices such that, every vertex in V has no neighbours in V, ie there exist an IS with $\geq k$ vertices then in the reduced graph G' we are sending the original graph G along with V extra vertices which are connected to each of the v' vertices ($|V| = |v'|$). Thus each of the vertices v' in G' now has a neighbours formed by the newly formed vertices ie the newly formed vertices will separately form a set s.t no 2 vertices from that set ~~have a~~ are neighbours of each other. and such vertices are $|V|$ in number. Also in the graph G' we have the original graph G, which can also form a ~~sep~~ separate group of vertices st no 2 vertices are neighbours of each other, and such vertices are $\geq |k|$ in numbers as in the original graph we had n vertices in the set V. Now when we combine the two sets of G' ~~(|k| + |V|)~~ they also form a valid set of vertices s.t every vertex have ~~or~~ atmost one neighbour (which are the newly added vertices) which satisfies the definition of ~~ACX~~ ACX, thus ACX will also return a Yes instance. The addition of one new vertex to each of the vertices in G' ensures that ~~not~~ whatever set of $\geq k$ vertices we were getting in ~~AC~~ IS will be present in ACX and along with that the newly connected components will be present in set of ACX. No other vertices can be present as now each vertex has atmost (in our case exactly) one neighbour. ie the newly connected component.

Let IS problem have graph $G(V,E)$ with k as the integer value. Now let k_s' be the ~~no~~ number of ~~IS~~ vertices we got in solution from IS algorithm. To return a yes instance $k_s' \geq k$ — ① must be satisfied.

now after reducing the graph we get $G'(2V, E)$ and integer $k \leq k+V$. from ① we can write

$k_s' \geq k$ where k_s' is the

$k_s' + V \geq k + V$ ~~solution~~ number of

$k_s'' \geq k+V$ vertices we got from

$k_s'' \geq k'$ AEX algorithm in its solution, which is

also ~~gr~~ atleast as much as k' ie the required/given number of vertices to be present in the solution set of AEX. Thus it satisfies the AEX problem and thus it also gives a ~~True~~ Yes instance.

If IS returns a NO instance ie there exist no IS with atleast k vertices, representing this mathematically →

G(V, E) with k integer of IS will give a NO instance when ks' (the ~~solut~~ number of vertices which forms the ~~soluti~~ independent set) < k ① ie we dont have an IS with atleast k vertices.

now reducing the graph with the help of reduction algorithm given above we have —

G' (2V, E) graph with k' = k+V ② ~~set~~ as the integer. Let ks'' be the solution of ACX algorithm ie the number of vertices formed with every vertex having atmost one neighbour.

now from eqn ① we can write —

$$ks' < k$$

$$\Rightarrow ks' + V < k+V$$

$$\Rightarrow ks'' < k' \quad [\, k+V = k' \text{ from } ② \,] \quad — ③$$

ks'' will be ~~the~~ ks' + V because V vertices are added as neighbours to the graph G' (one ~~to~~ new vertex to each existing vertex) which will themselves form an IS now the original graph G is also present as a part of G' which will have its own IS of ks'. Thus these 2 sets can be combined which will satisfy that each vertex has atmost one neighbour (the newly ~~form~~ added vertices) ~~and~~ which is nothing but ks''. ks'' will not contain other vertices as already the newly added vertices ~~neighbours of v~~ are satisfying the atmost one neighbour of v criteria and any new vertex added will violate this.

Thus from equation 3 we can see Aex ~~of~~ problem will return a NO as no set of atleast k' vertices ~~of~~ is formed. The solution set has ks'' vertices which is less than k'.