Clique Cover decision problem:

Given a Graph G = (V, E), and an integer k, is there a Clique Cover with k cliques or less?

Prove Clique Cover is NP:

Verification Problem:

Given a Graph G, a set of cliques C, and an integer k, verify that C is a Clique Cover of Graph G, with k cliques or less.

Verification Algorithm CCVerify(G, C, k):

Input: A Graph G = (V,E), and a certificate as a set of cliques C containing each vertex in this G, and an integer k.

Output: True if this certificate has k or less cliques, and it is a Clique Cover. False otherwise.

1: Count the number of cliques in C, if it is more than k, return False. This will take time k which is less or equal to |V|. It is Linear time.

2: Keep tracking of the number of times the Vertex in each clique c. Because the Vertex is in only one clique. If the vertex can be found in more than one clique, return False. This process will take linear time through iterating each Vertex in each clique c in C.

3: For each clique c in C, and for each vertex in c, verify that the vertex v is adjacent to other vertex in the same clique c. It can be done through examining each vertex’s adjacent list once, which will be in Linear time.

Prove Clique Cover is NP-Hard

3-colorability ≤p Clique Cover

Theorem: If 3 colorability cannot be decided in polynomial time, neither can Clique Cover.

If 3 colorability cannot be decided in polynomial time, neither can Clique Cover.

Proof by contradiction:

Assume clique cover can be solved in polynomial time. This means there is a polynomial time algorithm A that given a graph G and integer k, can decide if G has a clique cover with at most k cliques.

Prove Clique Cover is NP-Hard:

Theorem: If 3 colorability cannot be decided in Polynomial time, neither Clique Cover can.

Proof by contradiction:

Assume clique cover can be solved in polynomial time. This means there is polynomial time Algorithm A that given a Graph G and an integer k, can decide if G have a clique cover with k or less cliques.

Proof by contradiction:

Assume Clique Cover can be solved in polynomial time. This means that there is a polynomial time Algorithm A that Given a Graph G, and an integer k, can decide if G has a clique cover with k or less cliques.

Algorithm 3colorabilityToCC(G):

Input: A Graph G

Output: A Clique Cover of G

1: Construct a complement graph G’ of Graph G

2: Use Algorithm A to find the clique cover of the G’

3: If the clique cover has 3 sets or less, make all vertices in each set have the same color. And different sets use different colors to mark.

4: Return Clique Cover of G.

Prove Clique Cover is NP-Hard

Theorem: If a colorability cannot be decided in Polynomial time, neither Clique Cover can.

Proof by contradiction:

Assume Clique Cover can be solved in Polynomial time. This means that there is a Polynomial time Algo A that Given a Graph G, and an integer k can decide if G has a Clique Cover with k or less cliques.

Algorithm 3ColorabilityToCliqueCover(G):

Input: A graph G

Output: the clique cover of G

1: Construct the complement Graph G’ of G

2: Use Algorithm A to find the Clique Cover of G’

3: If the Clique Cover of G’ has 3 or less sets, then for each set, use the same color to mark vertex. Different sets use different colors to mark.

4: Return Clique Cover of G

Proof correctness of reduction in 2 parts (2 directions)

=>Part 1: If G is 3-colorable, then G’ has a clique cover with 3 sets or less. This means, if G is 3-colorable there does not exist an edge between 2 vertices u and v that have the same color. Thus, in G’, there will be an edge between u and v. That is, colors in G represent vertices that are in the same clique in G’. There are 3 colors for a

Proof correctness of reduction in 2 parts (2 directions)

Part 1: If G is 3 colorable, then G’ has a Clique Cover with k = 3 or less cliques.

G is 3 colorable. This means there does not exist an edge between u and v with same color in Graph G. While in G’, there is an edge between u and v with same color. So, each color in Graph G represents each clique cover in Graph G’. Totally 3 different colors in G, therefore, G’ has 3 cliques.

Proof the correctness of reduction 3 colorability ≤p Clique cover. 2 directions.

-> Part1: If Graph G is 3 colorable, then G’ has a Clique Cover with k = 3 sets or less.

Graph G is 3 colorability. It means that there is no edge between each vertex u and v with same color in G. In the complement of G as G’, there is an edge between each vertex u and v with same color in G’. Therefore, the same color vertex in G’ can be contained in one clique. 3 different colors in G’ will have 3 cliques. That’s to say: G’ has a Clique Cover with k = 3 sets or less.

<- Part2: If G’ has a clique cover with k =3 sets or less, then G is 3 colorable.

That means that V1, V2, V3 are 3 sets of Clique Cover in G’. Thus, there must be an edge between u and v in the same clique in G’. For Graph G, there is no any edge between u and v. Therefore, if u and v are assigned with same color and 3 cliques in G' will be provided 3 different colors, G will be 3 colorable and no conflict.

Theorem: If 3-colorability cannot be decided in polynomial time, neither the CC can.

Proof by contradiction:

Assume CC can be solved in polynomial time. This means there is a polynomial time Algorithm A that given a Graph G, and integer k, can decide that G has k or less cliques.

Next: we gave a reduction of 3 color <=p CC, and prove its correctness in both directions.

We assume that Algo A exists. However, if 3-colorability <=p CC, then we can use Algo A to decide 3 colorability. But 3-color is NP-C, there is no such Algorithm.

So, by contradiction, we proved CC does not have such an algorithm and it is NP-H.

Finally, we have proved CC is NP and NP-H, so CC is NP-C.