Homework 10

1. **(20pts) Given a graph G = (V, E) and two integers k, m, a clique is a subset of vertices such that every two distinct vertices in the subset are adjacent.**
   a) The Clique problem asks: Given a graph G, and a number k ≥ 0, does G have a clique of size k. Show that this problem is NP-complete.
   Hint: Reduce from Independent Set.
   Ans:
   Here, we reduce the independent set problem to prove that Clique problem is NP-Complete.
   Reduction:
   We know that an independent set in a graph is a subset of vertices where no two vertices in the subset are adjacent to each other. Hence, if we invert the edges of the graph to create a graph G', if there's no edge between vertices, there will be an edge between them. We set k' = k (here, k' is |V|-k).

   Claim:
   G has an independent set of size k if and only if G' has a clique of size k'.

   Proof:
   If G has an independent set of size k, then G' has a clique of size k':
   Since inverting the edges of G creates a complementary graph, any independent set in G corresponds to a clique in G'. Moreover, if there are k vertices in the independent set in G, there are k' vertices not in the independent set. These vertices form a clique in G'.

   If G' has a clique of size k', then G has an independent set of size k:
   By the construction of G', any clique of size k' in G' corresponds to a set of vertices in G with no edges between them. This set of vertices forms an independent set in G. Moreover, since the size of the clique in G' is k', the number of vertices not in the clique is |V| - k' = k. Therefore, this set of vertices forms an independent set of size k in G.

   This runs in polynomial time. Since independent set problem can be reduced to clique problem, therefore clique problem is also NP Complete.

   b) The Dense Subgraph Problem is to determine if given graph G, and numbers k, m ≥ 0, does there exist a subgraph G' = (V', E') of G, such that V' has at most k vertices and E' has at least m edges. Prove that the Dense Subgraph Problem is NP-Complete.
   Ans:
   Now that we have proved the clique problem to be NP Complete, we will reduce the clique problem to prove that Dense Subgraph problem is also NP Complete.
   Reduction:
   For the clique problem, if k' = k and m = k*(k-1)/2, then this will construct an instance for Dense Subgraph problem.

Claim:
G has a clique of size k if and only if G has a dense subgraph G' with at most k' vertices and at least m' edges.

Proof:
If G has a clique of size k, then G has a dense subgraph G':
Any clique of size k in G corresponds to a subgraph with k vertices, where each vertex is connected to every other vertex. This forms a dense subgraph with k vertices and k*(k-1)/2 edges.

If G has a dense subgraph G' with at most k' vertices and at least m' edges, then G has a clique of size k:
Since G' has at most k' vertices, there must exist a complete subgraph (clique) with k vertices in G'. Thus, G has a clique of size k.

This runs in polynomial time. Since Clique problem can be reduced to Dense Subgraph problem, therefore dense subgraph problem is also NP Complete.

2. **There are n courses at USC, each of them scheduled in multiple disjoint time intervals. For example, a course may require the time from 9am to 11am and 2pm to 3pm and 4pm to 5pm (you can assume that there is a fixed set of possible intervals). You want to know, given n courses with their respective intervals, and a number K, whether it's possible to take at least K courses with no two overlapping (two courses overlap if they have at least one common time slot). Prove that the problem is NP-complete. (20 points) Hint: Use a reduction from the Independent Set problem to show NP-hardness**
Ans:
Here, we reduce Independent Set problem to prove that this problem is NP Complete.
Reduction:
Let each vertex in graph G represent a course. For each edge, we assign a common time slot to the courses. These slots have to overlap so that we don't take both courses simultaneously without overlap.
Set K' = k.

Claim:
G has an independent set of size k if and only if it's possible to take at least K' courses with no two overlapping.

Proof:
If there exists an independent set of size k in G:
These vertices (courses) do not share any edges (time slots) with each other. Therefore, it's possible to take all these courses without any overlap. The size of this independent set is k, which satisfies the condition of taking at least K' courses.

If it's possible to take at least K' courses with no two overlapping:
These courses do not share any time slots with each other. Therefore, they form an independent set in the graph G. The number of courses we can take is at least K', which satisfies the condition of finding an independent set of size k.

This runs in polynomial time. Since independent set problem can be reduced to this problem, therefore this problem is also NP Complete.

3. **Consider the partial satisfiability problem, denoted as 3-Sat(α) defined with a fixed parameter α where 0 ≤ α ≤ 1. As input, we are given a collection of k clauses, each of which contains exactly three literals (i.e. the same input as the 3-SAT problem from lecture). The goal is to determine whether there is an assignment of true/false values to the literals such that at least αk clauses will be true. Note that for α = 1, we require all k clauses to be true, thus 3-Sat(1) is exactly the regular 3-SAT problem. Prove that 3-Sat(15/16)is NP-complete. (20 points)**
   Hint: If x, y, and z are variables, note that there are eight possible clauses containing them: (x ∨ y ∨ z),(!x ∨ y ∨ z),(x∨!y ∨ z),(x ∨ y∨!z),(!x∨!y ∨ z),(!x ∨ y∨!z),(x∨!y∨!z),(!x∨!y∨!z) Think about how many of these are true for a given assignment of x, y, and z.
   Ans:
   3SAT(15/16) is a NP problem as when we run the code, we can count the number of clauses that yield true or false. This runs in polynomial time.
   For NP Hard, we reduce the original 3SAT problem to prove that this problem is NP Complete.
   Reduction:
   For a 3SAT Problem, there are 8 possible clauses. For each set of clauses, we create 8 new clauses C' to 3SAT(15/16) and ensure that at least 15/16 of the original clauses can be satisfied. This will give us the total number of clauses as multiple of 8.

   This runs in polynomial time. Since 3SAT can be reduced to 3SAT(15/16), therefore, 3SAT(15/16) is also NP Complete.

4. **Consider the vertex cover problem that restricts the input graphs to be those where all vertices have even degree. Call this problem VC-EDG. Show that VC-EDG is NP-complete. (15pts)**
   Ans:
   Here, we reduce the Vertex Cover problem to prove that VC-EDG problem is NP Complete.
   Reduction:
   For each vertex in G, if the degree is odd, we add a new vertex v' connected to v by an edge. This ensures that vertices in G' have even degrees. Set k' = k.

Claim:
G has a vertex cover of size at most k if and only if there exists a vertex cover of size at most k' in G'.

Proof:

If there exists a vertex cover of size at most k in G:
We can select the same set of vertices in G' to cover all edges, excluding the new vertices introduced by us. Since each new vertex is connected to an original vertex, selecting an original vertex in the cover also covers its corresponding new vertex. Thus, the size of the vertex cover remains at most k'.

If there exists a vertex cover of size at most k' in G':
Since each new vertex introduced by us is connected to an original vertex, we can simply remove these new vertices from the cover to obtain a valid vertex cover for G with size at most k.

This runs in polynomial time. Since Vertex cover problem can be reduced to this problem, therefore VS-EDG problem is also NP Complete.