HW 10

Q1, a)

1- Prove that the clique problem in in NP.

Certificate: a subset of vertices that form a clique of size k.

Certifier: can easily check in polynomial time that:

a- The subset contains k vertices.

b- Every two distinct vertices in the subset are adjacent

A and B can be easily done in polynomial time.

→ Clique problem ∈ NP

2- choose 2ndepedent Set for our reduction.

3- will show that Independent Set $\leq_p$ Clique

We will start with an instance of the Independent Set problem (Is there an Independent set of size at least k in Graph G) and will construct a graph G' such that G has an Independent set of size at least k iff G' has a clique of size at least k.

Construction of G':

- Transform the graph G into its complement G^c.
- In G^c, two vertices are adjacent iff they are not adjacent in G.

Proof of correctness for the reduction step:

A- If G has an Independent set of size at least k, then G^c has a clique of size at lease k.

In the original graph G, an independent set of size k means there are k vertices none of which are directly connected by an edge.

In the complement graph G^c, these k vertices will all be mutually connected because we have an edge in G^c wherever there was no edge in G. Thus, if there is an independent set of size k in G, there is a clique of size k in G^c, and vice versa.

B- If G^c has a clique of size at least k, then G has an independent set of size at least k. The reason is that the vertices in the clique are adjacent to each other in G^c, so they must not be adjacent to each other in G, forming an independent set.

---

b)

1- Prove that the Dense Subgraph Problem is in NP.

Certificate: a subgraph G' of G.

Certifier: Can easily check in polynomial time that

a- G' has at most k vertices

b- G' has at least m edges

a and b can be easily done in polynomial time.

→ Dense Subgraph Problem $\in$ NP.

2- Choose clique for our reduction.

3- We will show that Clique $\leq_p$ Dense Subgraph Problem

We will start with an instance of the clique problem (Is there a clique of size at least k in graph G) and will construct an instance of the Dense Subgraph Problem such that G has a clique of size at least k iff G has a subgraph with at most k vertices and at least $m = \frac{k(k-1)}{2}$ edges.

Construction:

- Use the same graph G.
- Set the number of vertices in the dense subgraph problem to k.
- Set the number of edges in the dense subgraph problem to $m = \frac{k(k-1)}{2}$, which is the number of edges in a clique of size k.

Proof of correctness for the reduction step:

A - if G has a clique of size at least k, then G has a subgraph with at most k vertices and at least $m = \frac{k(k-1)}{2}$ edges. The reason is that a clique of size k is a subgraph with k vertices and $\frac{k(k-1)}{2}$ edges.

B - if G has a subgraph with at most k vertices and at least $m = \frac{k(k-1)}{2}$ edges, then G has a clique of size at least k. The reason is that for a subgraph to have at least $\frac{k(k-1)}{2}$ edges, it must have at least k vertices. And if a subgraph with k vertices has $\frac{k(k-1)}{2}$ edges, it means every two vertices in the subgraph are connected by an edge, forming a clique.

Q2

1- Prove that the course scheduling Problem is in NP.

Certificate: a set of at least k courses that do not overlap.

Certifier: Can easily check in polynomial time that

a- The set contains at least k courses.

b- No two courses in the set overlap (i.e. they do not have any common time slot)

a and b can be easily done in polynomial time

→ Course Scheduling Problem $\in NP$.

2- Choose Independent Set for our reduction.

3- Will show that Independent Set $\leq_p$ Course Scheduling Problem

We will start with an instance of the instance of the Independent Set problem (is there an Independent Set of size at least k in graph G) and will construct a set of courses and time slots such that there are at least k courses that do not overlap iff we have an dependent set of size at least k in G.

Construction of courses and time slots:

- For each vertex v in V, create a course C_v,
- Define a set of time intervals such that there is one interval for each vertex. Specifically, each course C_v will have a time interval that corresponds uniquely to vertex v.
- For each edge (u,v) in E, make the time intervals of courses C_u and C_v overlap. This can be achieved by assigning overlapping time slots to these courses based on their corresponding vertices being adjacent in the graph.

Proof of correctness for the reduction step:

A - If we have an independent set of size at least k in G, we can find at least k courses that do not overlap. The reason is that since the k nodes in G are independent, they do not share edges. Therefore, the courses corresponding to these k nodes will not have any time slots in common, i.e. they do not overlap.

B. - If we have at least k courses that do not overlap, we can find an independent set of size at least k in G. Since these courses do not have any time slots in common, the corresponding nodes in G will have no edges in common, or in other words, they will be independent and will form an independent set of size at least k.

Q3

1- Prove that 3-Sat(15/16) is in NP.

Certificate: a true value assignment

Certifier: can easily check in polinomial time that.

a- determine the number of satisfied clauses and proceed to make a comparison with 15K/16.

a can be easily done in polynomial time. → 3-Sat(15/16) ∈ NP

2- Choose 3-SAT for our reduction

3- Will show that 3-SAT $\leq_p$ 3-SAT(15/16).

For every group of 8 clauses in the 3-SAT problem (if the number of clauses is not a multiple of 8, adjustments can be made or irrelevant clauses added to meet the condition), add 8 new clauses composed of three entirely new variables, covering all possible combinations of these new variables.

A solution to the original 3-SAT problem exists iff a solution to the constructed 3-SAT(15/16) exists.

proof of correctness for the reduction step:

A- If the original 3-SAT problem is solvable, then at least 15/16 of the clauses in the newly constructed problem can be satisfied

The newly added 8 clauses are designed to cover all possible truth value combinations, ensuring that at least 7 clauses are true regardless of how values are assigned. Since only one clause is false under any truth assignment, this ensures that in a total of 16 clauses, at least 7 new clauses are true, thus at least 15 clauses are true, meeting the 15/16 requirement.

B- If at least 15/16 of the clauses in the new problem can be satisfied, the original problem must also be fully solved.

Because only 7 out of the 8 new clauses can be satisfied by any given assignment. Hence, if any of the original clauses were unsatisfied, it would be impossible to meet the 15/16 threshold.

Q4

1- Prove that Vertex Cover with only Even-degree vertices (VCE) is in NP.

Certificate: a set of vertices that form a vertex cover.

Certifier: can easily check in polynomial time that

a- The set of vertices from a vertex cover (i.e. the combination of all edges that are adjacent to the collection of vertices encompasses all edges in the graph.

a can be easily done in polinomial-time . → VCE ∈ NP.

> We can use the same certificates and verifiers as for the VC problem because whether the degree of a vertex in the graph is even or not does not affect whether the vertex set can cover all edges. Regardless of whether the degree of the vertices is odd or even, as long as the set of vertices covers all edges, it is a valid vertex cover.

2- choose Vertex Cover (VC) for our reduction

3- Will show that $VC \leq_p VCE$

We will start with an instance of the VC problem (is there an vertex cover of size at most k in G) and will construct G' such that G has a vertex cover of size at most k iff G' has a vertex cover of size at most k+2.

Construction of G': G' will have the same set of nodes and edges in G plus a number of new nodes and edges. The nodes in G' that exist in G will belong to the set S. We now introduce a new set of $\{v_1, v_2, v_3\}$ in G':

- we use $\{v_1, v_2, v_3\}$ to make a triangle $\{(v_1,v_2),(v_2,v_3),(v_3,v_1)\}$ and then connect v_1 to all vertices that have odd degrees in G.

The Reason: For each vertex in the original grap G, its degree is either already even or odd. To convert odd degree vertices to even degree, we connect each old degree vertex with an edge to the new vertex v_1. Moreover, the number of vertices with odd degrees in G should be even. Because, each edge will add 2 degrees. And the total degrees should be a even number.

- Thus, the degree of every vertex in G' is even.

Proof of correctness for the reduction step:

A- If we have a vertex cover of size at most k in G, we can have a vertex cover of size at most k+2 in G'.

Let S be defined as the vertex cover for G, the $\{v_1, v_2\}$ will cover the triangle $\{(v_1,v_2),(v_2,v_3),(v_1,v_3)\}$, where $\{v_1\}$ also covers the added edges between G and triangle. Thus. $S \cup \{v_1, v_2\}$ is a vertex cover for G'.

B- If we have a vertex cover of size at most k+2 in G', then we have a vertex cover of size at most k in G.

Let S' be the vertex cover of G'. S' must have at least two triangle vertices to cover the added triangles. By removing triangle vertices, the remaining part will be the original graph G. The vertex cover of G will be obtained from S' by subtracting 2.