Homework 10

Q1) (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 $k \geq 0$, does G have a clique of size k. Show that this problem is NP-complete. Hint: Reduce from Independent Set.

b) The Dense Subgraph Problem is to determine if given graph G, and numbers $k, m \geq 0$, 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.

Solution: Proving these problems are in NP, is trivial, so here we focus on proving their NP-hardness.
First, we prove that the Independent set problem $\leq_p$ Clique $\leq_p$ Dense Subgraph Problem. Given a graph G(V, E) and an integer k, an independent set decision problem outputs yes, if the graph contains an independent set of size k. For an arbitrary graph G = (V, E) of n vertices, we first get the complementary graph Gc of G. Gc contains an edge between two vertices if there was no edge between them in G, and vice versa. Thus, an independent set in G, i.e., one where each pair of vertices did not have an edge, becomes a clique in Gc and vice versa. This proves that Gc has a clique of size k if and only if G has an independent set of size k. This shows Independent set problem $\leq_p$ Clique.

Further, note that a clique of size k will always contain $k(k-1)/2$ edges. Thus, if we set m to $k(k-1)/2$ then, we can say that a graph has a clique of size k if and only if the same graph has a dense subgraph with at most k nodes and at least m edges (prove both directions of the statement clearly for completeness). This shows that Clique $\leq_p$ Dense Subgraph Problem.

Example: Consider a graph G = (V, E) where V = {1, 2, 3, 4} and E = {(1, 2),(2, 3)}. Suppose we want to determine if there's an independent set in G of size k=3. We first generate Gc, where Ec = {(1, 3),(1, 4),(2, 4),(3, 4}. Now, V′ = {1, 3, 4} is an independent set of size 3 in G, and a clique of size 3 in Gc. Further, if we set k = 4 and $m = 4 \cdot 3/2 = 6$, then there's a clique of size 4 if and only if there is a dense subgraph with 4 vertices and 6 edges (the answer is no to both here).

Rubric (20 pts)

• 3 pt: state the problem is in NP and mention how to verify it in polynomial time.
• 5 pt: Choose an NP-complete problem and say that we need to prove that chosen problem ≤p Dense Subgraph problem. Other reduction can also get 3 points. Note: wrong direction will get 0.
• 12 pt: Showing the polynomial reduction with details and its proof including:
4 pt: instance construction 8 pt: proving the claim. Each direction is 4 pts.

Q2) 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

Solution:
(a) (Showing Problem in NP) The solution of the problem can be verified in polynomial time (just check the number of the courses in the solution is larger or equal to K, and they don't have time overlap), thus it is in NP.
(b) (Showing Problem in NP-Hard) Given an independent set problem, suppose the graph has n nodes and asks if it has an independent set of size at least K. We map the instance of IS to an instance of course choosing problem by constructing a course Cu for each vertex u, a time interval Te for each edge e and letting a course Cu to consist of all intervals Te such that e is incident on u. Then vertices u and v are disjoint if and only if courses Cu and Cv are non-overlapping. Thus, there is an independent set of size k if and only if we can choose k courses with no two overlapping. Thus we can reduce the independent set problem to the course choosing problem in polynomial time. Since the independent set problem is NP-Complete, the course choosing problem is in NP-Hard.
Thus the course choosing problem is NP-Complete.

Rubric (20 pts)

- 5 pts: Proving course-choosing problem is in NP
- 15 pts: Proving course-choosing-problem is in NP-Hard

– 8 pts: the construction + explanation (can be any other constructions)

– 6 pts: the proof

– 1 pt: the conclusion

Q3: Consider the partial satisfiability problem, denoted as 3-Sat(α) defined with a fixed parameter α where $0 \le \alpha \le 1$. 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.

Solution:
To prove it's in NP: given a truth value assignment, we can count how many clauses are satisfied and compare it to 15k/16.
To prove it's NP-hard:
We will show that 3-SAT ≤p 3-SAT(15/16). For each set of 8 original clauses, we add 3 NEW VARIABLES, and create all the 8 possible clauses on these 3 new variables. If the number of clauses is a multiple of 8, then we have created m new clauses for the m existing ones. Now, any assignment will satisfy only 7/8 of the new clauses by construction, so we can say, all of the original clauses in a valid solution can be satisfied if and only if 15/16 of all the new clauses can be satisfied (convince yourself by proving the previous statement both ways). Next, if the number of clauses is not a multiple of 8, say it is of the form 8a+b with b < 8. Then we follow the procedure above for the first a groups of 8 clauses and end, leading to a total of 16a + b clauses in the new instance. Remember that regardless of any assignment, exactly 7a of the new clauses can be satisfied. Then, it can be shown that one can achieve the factor of 15/16 satisfiability in the new instance if and only if all 8a + b of the original ones can be satisfied (using b¡8 and a bit of algebra).
So, 3-Sat(15/16) is in NP and is NP-hard which makes it NP-Complete.

Example: original given formula is

(a ∨ b ∨ c) ∧ (!a ∨ b ∨ c) ∧ (a ∨ !b ∨ c) ∧ (a ∨ b ∨ !c) ∧ (!a ∨ !b ∨ c) ∧ (!a ∨ b ∨ !c) ∧ (a ∨ !b ∨ !c) ∧ (!a ∨ !b ∨ !c)
So we add our 8 new clauses so that the formula now contains a total of 16 clauses, i.e.:
(a ∨ b ∨ c) ∧ (!a ∨ b ∨ c) ∧ (a ∨ !b ∨ c) ∧ (a ∨ b ∨ !c) ∧ (!a ∨ !b ∨ c) ∧ (!a ∨ b ∨ !c) ∧ (a ∨ !b ∨ !c) ∧ (!a ∨ !b ∨ !c) ∧ (d ∨ e ∨ f) ∧ (g ∨ h ∨ i) ∧ (!a ∨ !h ∨ !i) ∧ (x ∨ y ∨ z) ∧ (!x ∨ y ∨ z) ∧ (x ∨ !y ∨ z) ∧ (x ∨ y ∨ !z) ∧ (!x ∨ !y ∨ !z)
If the original 8 can be satisfied with some assignment, then additionally setting any assignment to the new variables satisfies 7 of the new clauses, thus 15/16 of them in the new instance. If the original number of clauses in a formula is not a multiple of 8, say it was 13, then we still only add the above 8 new clauses making a total of 21. 15/16 fraction of that is $19\frac{11}{16}$, so satisfying 19 clauses doesn't get us above the required ratio, and we must satisfy 20. Since exactly 7 of the newly added 8 can be satisfied, the total of 20 can be achieved if and only if all the original 13 can be satisfied.

Rubric:

- 5 pt: state the problem is in NP and mention how to verify it in polynomial time by checking how many clauses are satisfied.
- 5 pt: Choose an NP-complete problem and say that we need to prove that chosen problem ≤p 3-SAT (15/16). Other reduction can also get 3 points. Note: wrong direction will get 0.
- 5 pt: instance construction for showing the polynomial reduction with details
- 5 pt: using the construction proof the polynomial reduction.

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)

Solution:
First we need to prove even degree vertex cover problem is in NP.
The certifier takes a subset of vertices as its certificate. It verifies that the subset is of size at most K and each of the original graph's edges has one of its endpoints in the given subset. Thus problem is in NP (the same certifier to the original vertex cover problem).

Then we need to prove VC-EDG is NP-Hard.

We claim that it is polynomial time reducible from the original vertex cover problem. Let (G = (V, E), k) to be an input instance of Vertex Cover. Because each edge in E contributes a count of 1 to the degree of each of the vertices with which it connects, the sum of the degrees of the vertices is exactly 2|E|, an even number. Hence, there is an even number of vertices in G that have odd degrees.

Let U be the subset of vertices with odd degrees in G.
Construct a new instance ($\bar{G}$ = (V0, E0), k + 2) of Vertex Cover, where V0 = V ∪ {x, y, z} and E0 = E ∪ {(x, y),(y, z),(z, x)} ∪ {(x, v)|v ∈ U}. That is, we make a triangle with three new vertices, and then connect one of them (say x) to all the vertices in U. The degree of every vertex in V0 is now even. Since a vertex cover for a triangle is of (minimum) size 2, it can be shown that $\bar{G}$ has a vertex cover of size at most k + 2 if and only if G has a vertex cover of size at most k.

Hence, vertex cover with only even degree vertices is NP Complete.

Rubric (15 pts)

- 6 pt: state the problem is in NP and mention how to verify it in polynomial time by checking if it is a valid vertex cover with even degree.
- 3 pt: Choose an NP-complete problem and say that we need to prove that chosen problem ≤p even degree vertex cover. Other reduction can also get 3 points. Note: wrong direction will get 0.
- 6 pt: Showing the polynomial reduction with details and its proof.

UNGRADED PROBLEMS

Q5: Let S be an NP-complete problem, and Q and R be two problems whose classification is unknown (i.e. we don't know whether they are in NP, or NP-hard, etc.). We do know that Q is polynomial time reducible to S and S is polynomial time reducible to R. Mark the following statements True or False based only on the given information, and explain why.

(i) Q is NP-complete (4pts)

Sol: False. Because Q <p S, Q is at most as hard as S. Because S is in NPC, Q is not necessary to be in NPC, e.g., it could be in P, or could be in NP but not in NPC.

(ii) Q is NP-hard (4pts)

False. Because Q <p S, Q is at most as hard as S. Then Q is not necessary to be in NP-hard, e.g., it could be in P.

(iii) R is NP-complete (4pts)

False. Because S <p R, R is at least as hard as S. Then R is not necessary to be NPC. It is possible to be in NP-hard but not in NPC.

(iv) R is NP-hard (4pts)

True. Because S <p R, R is at least as hard as S. Then R in NP-hard.

Rubrics:
+2pts: If correctly stated T/F
+2pts: correct explanation