SECTION 10 SOLUTIONS

CS 121 Fall 2020

Eric Lin

1. Show that for every $F, G, H : \{0,1\}^* \to \{0,1\}$, if $F \leq_p G$ and $G \leq_p H$, then $F \leq_p H$.

Solution to 1:

By definition, we have polynomial time computable functions R_1 and R_2 such that $F(x) = G(R_1(x))$ and $G(x) = H(R_2(x))$. Then, $F(x) = H(R_2(R_1(x)))$. We know that since this is a polynomial time reduction, $R_1, R_2 \in$ P, and P is closed under composition. This means that $R_2(R_1(x))$ is computable in polynomial time.

2. Given an undirected graph $G = (V, E)$, a vertex-cover is a subset $V' \subseteq V$ s.t. for all $(v_1, v_2) \in E$, either $v_1 \in V'$ or $v_2 \in V'$. Consider the function VERTEX-COVER$(G, k) = 1$ iff G has a vertex cover of size k, and 0 otherwise. Is VERTEX-COVER NP-complete? Prove it or show why not.

Proof. We first have to decide if VERTEX-COVER is NP-complete or not. This decision largely lies on whether it is NP-hard. We will follow a similar approach as how we proved CLIQUE was NP-complete in section.

First, let's establish that VERTEX-COVER $\in$ NP.

Let a certificate w be a subset $V' \subseteq V$ which contains vertices in a vertex cover. We then verify this certificate by checking if every edge in the graph is connected to a vertex in V'. This is simply done by looping through every edge and at each iteration checking if at least one endpoint of the edge is a vertex in V'. The for loop takes a polynomial amount of time with respect to the number of edges and vertices. Finally, we check that $|V'| \geq k$.

Below are two different solutions in how we can prove that VERTEX-COVER is NP-hard.

Solution version 1 – reduction from 3SAT: We seek to create a graph representing the 3SAT equation such that it's satisfiable if and only if there is a vertex cover of size $n + 2m$, where n is the number of literals in the 3SAT equation and m is the number of clauses.

Consider the following construction of such a graph. For each variable x in the formula, add 2 nodes x and $\bar{x}$ and connect them with an edge, kind of like a dumbbell. For each clause x_i, x_j, x_k, add a new node that represents each of the literals in that clause, and connect those

Date: November 16, 2020.

new nodes together in a triangle so that each literal is connected to the others in the same clause. Then, connect each of the new literal nodes to the original variable nodes.

We posit that this graph has a vertex cover of size $n+2m$ if and only if the 3SAT expression it represents is satisfiable. We see first that if there is a satisfiable solution, we can construct a vertex cover from it. We do so by selecting the n variable nodes (1/2 of each pair of the dumbbell nodes). This will cover each of the edges connecting the variable nodes, and 1 out of each of the three nodes connecting the variable nodes with the literal nodes (the cross edges between dumbbells and triangles). Now, to cover the remaining cross edges and the edges between literal nodes in each of the triangles, you will have to select up to 2 nodes for each of the clauses, which is up to $2m$ overall. Thus, there will be a vertex cover of at most $n + 2m$ nodes if the 3SAT expression is satisfiable.

To show the other direction, we see that given a vertex cover of size $n + 2m$, we must have an assignment that satisfies the expression. n of the vertices must be used to cover the variable nodes, since there are n edges there and at least one of their endpoints must be covered. Then, the remaining 2m vertices must cover the edges within each triangle, as well as the cross edges between the triangles and the variable nodes, and so we have a satisfying assignment. Finally, we need to show that the transformation from the expression to the graph takes polynomial time. There are exactly $2n + 3m$ vertices, and there are n edges to connect the variable nodes, $3m$ cross edges, and $3m$ nodes in all of the triangles, so that gives us both $O(n + m)$ vertices and edges. Thus, we have a polynomial time transformation.

Alternate solution – reduction from CLIQUE: Note that VERTEX-COVER can also be proved to be NP-hard if we reduce from CLIQUE. This is similar to how we reduced ISET to CLIQUE. We want to transform the inputs of CLIQUE to the inputs of VERTEX-COVER. Our reduction function takes in a graph G and generates its complement G'. We notice that a clique of size k in G is equivalent to a vertex cover set of size $|V| - k$ in G'. Then, in our reduction we change G to G' (complement) and k to $k' = |V| - k$. Thus, this reduction transforms the inputs of CLIQUE to VERTEX-COVER. This reduction can be computed in polynomial time by iterating through all the vertices and edges and performing a constant number of steps for each. A proof of completeness and soundness follows similar to our proof of CLIQUE in section.

□

3. Given n sets $S_1, S_2, \ldots, S_n$ such that

$$\bigcup_{i=1}^{n} S_i = A$$

the set cover of size k over these sets is a collection C of k of these sets such that

$$\bigcup_{i \in C} S_i = A$$

Given a collection of sets and an integer k, SET-COVER returns if there exists a valid set cover of a most size k over the given collection of sets. Prove that SET-COVER is NP-complete.

Proof. We know that SET-COVER is in NP because we can use a set of sets as the certificate, and can verify by checking that each element is a member of at least one set.

We now reduce from VERTEX-COVER. Suppose we have an instance of VERTEX-COVER (so a graph and a number k). Label the edges in the graph from 1 to m. For each vertex v create a set S_v that is composed of the edges of which v is a part. This transformation is polynomial time since it must run over the edges and then runs over the vertices (going over each edge twice more).

First suppose that there is a valid VERTEX-COVER. We claim that the sets associated with the vertices in the VERTEX-COVER (call these vertices V') form a valid SET-COVER. For any number associated with an edge $e = (u, v)$, either $u \in V'$ or $v \in V'$, which implies the number associated with e is either in S_u or S_v.

Now suppose there is a valid SET-COVER of size k. We claim the vertices associated with the sets in this SET-COVER (denote this set of vertices V') form a VERTEX-COVER. Suppose towards a contradiction there was an edge $(u, v) \in E$ such that $u, v \notin V'$. This implies the number associated with that edge would not be in the SET-COVER, so it would not be a SET-COVER, hence a contradiction. □

4. For each of the following, say whether the problem is in P, NP, is uncomputable, or whether we don't know.

(1) Given an integer x, determine if x has a prime factor that is at most k.
(2) Given an undirected graph graph, determine whether it is possible to partition its vertices into two sets, with at least k edges crossing between sets.
(3) Given a program Q, an input x, and a string 1^t, determine whether Q halts on x within t steps.

Solution to 4:

(1) in P (can use a modified Sieve of Eratosthenes that runs in polynomial time)
(2) in NP (the certificate is the partition)
(3) in P (the program can be simulated in polynomial time)