Homework #9

You do not need to turn in these problems. The goal is to reinforce what we learned in class, as well as to cover material we didn't have time to cover in class. The in-class quiz that will cover the same or similar problems. Material on homework can also appear on exams.

Some known $\mathcal{NP}$-complete problems

You can assume the following problems are $\mathcal{NP}$-complete, and reduce them to the problems below to show that the problems below are also $\mathcal{NP}$-complete. Remember, to prove that problem X is $\mathcal{NP}$-complete, you must: (1) show that it is in $\mathcal{NP}$ (show that a solution can be verified efficiently) and (2) show that a known $\mathcal{NP}$-complete problem polynomial-time reduces to problem X.

3-**colorability.** *Given a graph G, can the set of vertices be partitioned into 3 sets such that no two vertices within the same set have an edge between them? In other words, can the graph be colored with three colors such that no adjacent vertices share a color?*

Vertex Cover. *Given a graph G and a number k, does G contain a vertex cover of size at most k? (Recall that a vertex cover $V' \subseteq V$ is a set of vertices such that every edge $e \in E$ has at least one of its endpoints in V'.)*

Subset Sum. *Given a set of n integers $S = \{x_1, x_2, \ldots, x_n\}$ and a target sum T, does there exist a subset $S' \subseteq S$ such that $\sum_{x \in S'} x = T$?)*

Problem 1: Approximate Subset Sum

Show that the following problem is $\mathcal{NP}-$Complete:
Given set S of n integers, $S = \{x_1, ..., x_n\}$, and two integers T, e such that $T > e \geq 1$ is there a subset $S' \subseteq S$ such that $T - e \leq \sum_{x \in S'} x \leq T$?

Solution

Again, we can see that this problem is in NP since given the appropriate set S we can check in polynomial time that it is a solution.
As the title suggests we will reduce Subset Sum to this problem. Let $A_1 = \{w_1, ..., w_n\}$ be the instance of Subset Sum and W the target sum. We set $A = \{2w_1, ..., 2w_n\}$, $B = 2W$ and $x = 1$. Finding a subset $S' \subseteq A_1$ such that $w(S') = W$ is equivalent of finding a subset $S \subseteq A$ such that $2W - 1 \leq w(S) \leq 2W \Rightarrow w(S) = 2W$ completing the reduction.

Problem 2: Efficient Recruiting

Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who is skilled at each of the n sports covered by the camp (baseball, volleyball, etc.). They have received job applications from

m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the **Efficient Recruiting** Problem. Show that **Efficient Recruiting** is $\mathcal{NP}$-Complete.

Solution

Efficient Recruiting is in NP, since given a set of k counselors, we can check that they cover all of the sports.

Suppose we had an algorithm A that solves *Efficient Recruiting*; here is how we would solve an instance of *Vertex Cover*. Given a graph $G = (V, E)$ and an integer k, we would define a sport S_e for each edge e and a counselor C_v for each vertex v. C_v is qualified in sport S_e if and only if e has an endpoint equal to v.

Now if there are k counselors that, together, are qualified in all sports, the corresponding vertices in G have the property that each edge has an end in at least one of them; so they define a vertex cover of size k. Conversely, if there is a vertex cover of size k, then this set of counselors has the property that each sport is contained in the list of qualifications of at least one of them.

Thus, G has a vertex cover of size at most k if and only if the instance of *Efficient Recruiting* that we create can be solved with at most k counselors. Moreover, the instance of *Efficient Recruiting* has size polynomial in the size of G. Thus if we could determine the answer to the *Efficient Recruiting* instance in polynomial time, we could also solve the instance of *Vertex Cover* in polynomial time.

Problem 3: Feedback Vertex Set, Directed

A set of vertices $S \subseteq V$ of a graph $G(E, V)$ is a feedback vertex set if the induced subgraph after deleting the vertices of S (and the respective edges) has no cycle. Show that the following problem is $\mathcal{NP}-$Complete:

FVS. *Given a directed graph $G(V, E)$ and non negative integer k, is there a Feedback Vertex Set(FVS) $S \subseteq V$ with k or less vertices?*

Solution

We claim that FVS is $\mathcal{NP}-$complete. FVS is in the class $\mathcal{NP}$ because a certificate would be the set of k vertices to remove. We would remove these vertices from the graph G to build a new graph G' . We would then run depth-first search from each vertex to test for cycles. This clearly takes polynomial time.

Next, we show FVS is $\mathcal{NP}-$hard by a reduction from Vertex Cover (VC). An instance of VC is an undirected graph G and a positive integer k. In our reduction, we build a new graph G' with the same vertices as G but with each edge (u, v) of G replaced by a pair of oppositely directed edges (u, v) and (v, u). Call this new directed graph G' . Our instance of FVS is G' , k (we use the same k as in the instance of VC).

Clearly this reduction takes polynomial time since it simply copies G but replaces each edge with two edges. Now, we show that there is a vertex cover of size k in G if and only if there is a subset of k vertices in G' whose removal breaks all cycles. First, assume that there is a vertex cover of size k in G. Now, remove these k vertices from G' along with the edges incident upon them. Notice that for any directed edge $(u, v) \in G'$, at least one of u or v must have been removed, because one of u or v must have been in our vertex cover. Thus, after removing these k vertices and their incident edges from G' , no two vertices have an edge between them, and consequently there can be no cycles. Conversely, assume that there exists a set S of k vertices in G' whose removal breaks all cycles. By construction every pair of vertices u, v that had an edge between them in G have a cycle $(u, v), (v, u)$ in G' . Since the removal of set S broke all cycles in G' , for each edge $(u, v) \in G$, the set S must contain either u or v (or both). Thus, S is a vertex cover of G.

Problem 4: 2018-colorability

We can extend the definition of 3-**colorability** to the following:

k-**colorability**: *Given a graph G, can the set of vertices be partitioned into k sets such that no two vertices within the same set have an edge between them?*

Show that 2018-**colorability** is $\mathcal{NP}$-complete.

Solution

First, note that for all k, k-**colorability** is in $\mathcal{NP}$. A colored graph is the certificate; it has polynomial size $(|V| + |E|)$. To check if the coloring is correct, check each edge and ensure that its incident vertices have different colors.

Next, we will show how to solve an instance of 3-**colorability** using 2018-**colorability**. That is, 3-**colorability** $\leq_p$ 2018-**colorability**. We will show that k-**colorability** $\leq_p$ $k+1$-**colorability**. Since $\leq_p$ is transitive, this argument (along with the membership in $\mathcal{NP}$ argument shown above) will show that k-**colorability** is $\mathcal{NP}$-complete for all $k > 3$, which includes 2018-**colorability**. Given an instance (graph G) of k-**colorability**, we will construct an instance of $k+1$-**colorability**. Take G, and construct G' by adding one vertex v. Add an edge between v and every other vertex in the graph. In a solution to $k+1$-**colorability**, the vertex v must be colored with some color. Note that in a correct solution, no other vertex in G' may use that color, since the vertex v has an edge to all other nodes. Then the remaining vertices must be colored using only the remaining k colors. This is possible if and only if the original graph G is colorable using k colors. Thus we have shown k-**colorability** $\leq_p$ $k+1$-**colorability**.

Note: You can also prove this directly by showing that 3-**colorability** $\leq_p$ 2018-**colorability**; given an instance of 3-**colorability**, add the complete graph on 2015 vertices, and add an edge from every new vertex to every old vertex. This uses the fact the complete graph on k vertices requires exactly k colors. In fact, the graph generated by applying the k to $k+1$ construction repeatedly to reach 2018 gives the same graph as adding the complete graph on 2015 vertices.