Practice Problems 11

Problem 1. In this problem we will explore some examples of efficient verification algorithms. For each of the decision problems listed below, show that the problem is in NP by presenting a polynomial time verification algorithm. In each case, explain what the certificate is, and present a short description of your verification algorithm. (If the algorithm is not too technical, this description can be given in English.) Recall that you only need to verify instances where the answer to the decision problem is "yes."

(**Hint:** Be careful. The efficient verification algorithm is not obvious in all cases. Note that these problems are not necessarily NP-complete.)

(a) k-Hamiltonian Cycle (k-HC): You are given an undirected graph $G = (V, E)$ and an integer k. Do there exist k vertex-disjoint simple cycles in G that include all the vertices of G? (See Fig. 1(a).) For this problem, assume that a cycle must traverse at least three distinct vertices and cannot repeat any vertex, except the first and last.

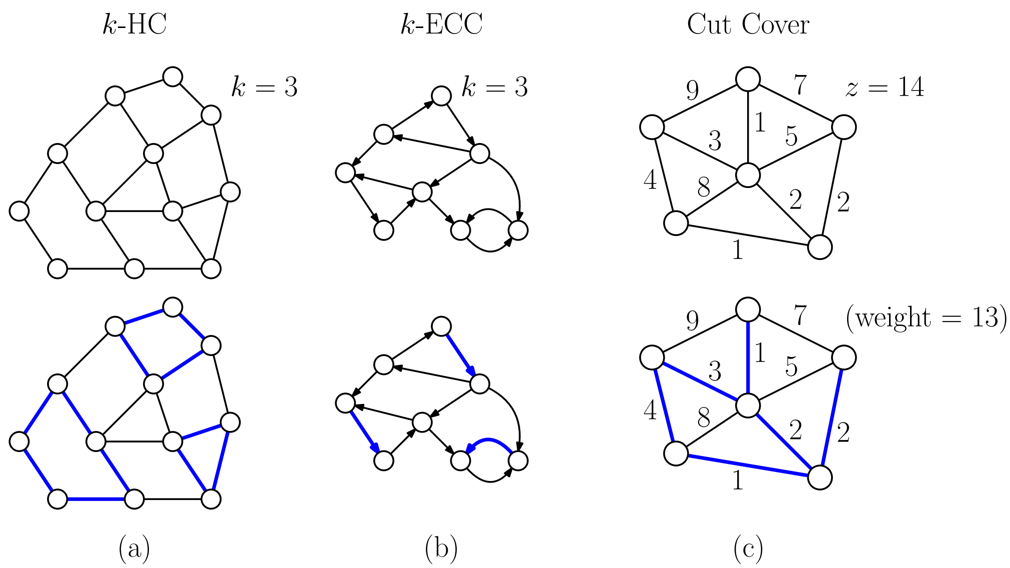

Figure 1: Verification algorithms.

(b) k-Edge Cycle Cover (k-ECC): Given a directed graph $G = (V, E)$ and a positive integer k, does there exist a subset $E' \subseteq E$ of size k such that every cycle in G passes through at least one edge of E'? (See Fig. 1(b).)

(c) Cut Cover: Consider a positively edge-weighted undirected graph $G = (V, E)$, where $w(u, v)$ denotes the weight on edge (u, v). A *cut* in such a graph is a partition of the vertex set $V = X \cup Y$, such that both X and Y are nonempty. An edge of E is said to *cross* the cut if one endpoint is in X and the other is in Y.

The Cut Cover problem is as follows: Given such a graph and a positive integer z, does there exist a subset $E' \subseteq E$ of total weight at most z such that for every cut in G at least one edge of E' crosses the cut? (See Fig. 1(c).)

Problem 2. The *Vertex-Cover* decision problem (VC) is: given a graph $G = (V, E)$ and an integer k, where $0 \leq k \leq |V|$, does there exist a subset $V' \subseteq V$ of size k such that every edge of G has at least one endpoint in V'. (For example, the graph in Fig. 2 has a vertex cover of size 4.) This topic will be discussed in Lecture 18, but this question can be answered without knowing the material from that lecture.

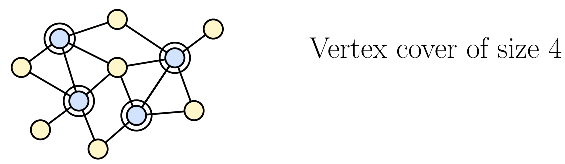

Figure 2: Vertex cover decision problem (VC).

Suppose that you had an oracle that can answer VC queries in polynomial time. (This is highly unlikely, since VC is NP-complete.) That is, given a pair (G, k), this oracle runs in polynomial time and returns "yes" or "no" depending on whether G has a vertex cover of size k. We want to convert this decision algorithm into an optimization algorithm.

(a) Explain how to use this oracle to determine the size k^* of the smallest vertex cover in G and further to identify which vertices are in the vertex cover. If there are multiple vertex covers of size k^*, your algorithm can return any of them. The number of oracle calls should be polynomial in the size of the graph. (Can you make it logarithmic in the size of the graph?)

(b) Provide a clear justification of why your procedure is correct and show that its running time is polynomial. (**Hint:** Be careful that your solution works even if the graph has multiple vertex covers of equal size.)

Problem 3. In this problem, we will consider more examples of using a black-box solution to a decision problem to compute the desired structure.

(a) **Hamiltonian Cycle:** Given an undirected graph $G = (V, E)$, does there exist a cycle that visits every vertex of graph exactly once?

Suppose that we had a function $\mathsf{HamCycle}(G)$, which (by some miracle) ran in polynomial time and returns `true` if G has a Hamiltonian cycle and `false` otherwise. Show that if G has a Hamiltonian cycle, then it is possible to use this function (as a black box) to compute the sequence of vertices on the Hamiltonian cycle in polynomial time.

(Let $n = |V|$ and $m = |E|$. For full credit, solve this problem using $O(m)$ calls to the function. For partial credit, any polynomial number of calls is allowed.)

(b) **3-Colorable:** Given an undirected graph $G = (V, E)$ can the vertices of G be labeled with three colors (say, 1, 2, and 3) such that no edge is incident to two vertices of the same color?

Suppose that we had a function 3Color(G), which (by some miracle) ran in polynomial time and returns true if G is 3-colorable and false otherwise. Show that if G is 3-colorable, then it is possible to use this function (as a black box) to determine the assignment of colors to the vertices.

(Let $n = |V|$ and $m = |E|$. For full credit, solve this problem using $O(n)$ calls to the function. For partial credit, any polynomial number of calls is allowed.)

Problem 4. In ancient times, King Arthur had a large round table around which all the knights would sit. Unfortunately, some knights dislike each other, cannot be seated next to each other. There are n knights altogether, $\{v_1, \ldots, v_n\}$, and the king has given you a list of pairs of the form $\{v_i, v_j\}$, which indicates that knights v_i and v_j dislike each other.

You are asked to write a program to determine if it is possible to seat the knights about the round table, called the *angry knight seating problem* (AKS). Show that AKS is NP-complete by proving

(a) AKS $\in$ NP

(b) UHC $\leq_P$ AKS, where UHC is the problem of determining whether an undirected graph has a Hamiltonian cycle.

Problem 5. Recall that an *independent set* in a graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices such that for all $u, v \in V'$, $\{u, v\} \notin E$. Intuitively, one might expect that the vertices of an independent set should have low degree. In this question, we will show that finding independent sets is hard, even if the degrees are required to be high.

In the *High-Degree Independent Set problem* (HDIS), you are given an undirected graph $G = (V, E)$ and a positive integer k, and you want to know whether there exists an independent set V' of size k such that each vertex of V' is of degree at least k in G. Letting $n = |V|$, you may assume that $1 \leq k \leq n - 1$, since otherwise the answer is trivially no.

For example, the graph in Fig. 3 has an HDIS for $k = 3$, shown as the shaded vertices. (It does *not* have an HDIS for $k = 4$, since although adding the topmost vertex would yield an independent set of size 4, there are two vertices in this independent set whose degrees are smaller than 4.)

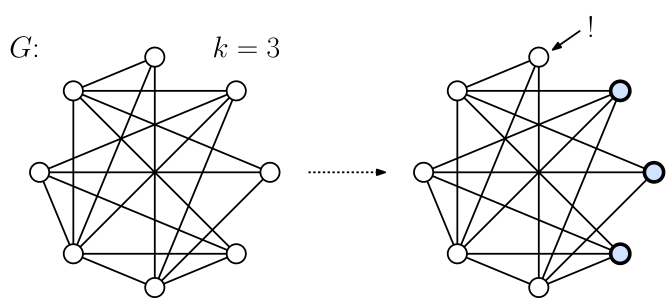

Figure 3: This graph has an HDIS for $k = 3$, but not for $k = 4$.

Show that HDIS is NP-complete by proving:

(a) HDIS $\in$ NP

(b) IS $\leq_P$ HDIS, where IS is the standard independent set problem. (**Hint:** Given an instance of IS, modify the graph to boost the degrees of all the vertices up to at least k, but without destroying any existing independent sets and not introducing any new independent sets.)