CS140: Algorithms

Spring 2001

Solutions to Assignment 10

1. In the study of NP-completeness we focus on "decision problems," which differ from standard search problems in that we are only required to answer a yes/no question about the input. In practice, of course, we usually want to solve the search variations of these decision problems. In this exercise you'll show that the search variations of several decision problems are really no harder than the decision version. You'll do this by reducing the search problem to the decision problem. Your reductions should be polynomial-time.

Dominating Set: A dominating set of a graph G is a subset of vertices W such that every vertex of G is either in W or is adjacent to a vertex in W.

Input: A graph G and integer k

Question: Does G have a dominating set of size at most k/

Input: A graph G

Search Output: A smallest dominating set of G

Use the decision algorithm on inputs G, K for $K = 1, 2, ...$ to determine the size, M, of the smallest dominating set of G. Now choose an unmarked vertex v of G. (Initially all vertices are unmarked.) Add a new vertex X attached by an edge to v. Ask the decision algorithm if the modified graph has a dominating set of size M. If yes, then mark v as a member of the dominating set and leave X in the graph. If no, mark v as a non-member and remove the vertex X. Repeat until M vertices are marked as members of the dominating set.

Clique: A clique in a graph G is a subset of vertices W such that each pair of vertices in W is connected by an edge in G.

Input: A graph G and an integer k

Question: Does G have a clique of size at least k?

Input: A graph G

Search Output: A largest clique of G

Use the decision algorithm on inputs G, K for $K = 1, 2, ...$ to determine the size, M, of the largest clique of G. Now choose an unmarked vertex v of G and eliminate it. Ask the decision algorithm if the modified graph has a clique of size M. If no, then reinstate v and mark it as a member of the clique. Repeat until M vertices are marked.

Hamiltonian Cycle: A Hamiltonian cycle of a graph G is a simple cycle that includes every vertex.

Input: A graph G

Question: Does G contain a Hamiltonian cycle?

Input: A graph G

Search Output: A Hamiltonian cycle of G if one exists and otherwise "No Hamiltonian cycle"

Use the decision algorithm to determine whether or not a Hamiltonian cycle exists. If not, then answer accordingly. If so, then choose an unmarked edge e of G and eliminate it. Ask the decision algorithm if the resulting graph has a Hamiltonian cycle. If not, reinstate e and mark it as a member of the Hamiltonian cycle. Repeat until n edges are marked, where $n = ||V||$.

3-coloring: A 3-coloring of graph G is an assignment of a unique color in {red, blue, green } to each vertex of G so that the endpoints of each edge are colored differently.

Input: A graph G

Question: Can G be 3-colored?

Input: A graph G

Search Output: A 3-coloring of G if one exists and otherwise "No 3-coloring"

Use the decision algorithm to determine if the graph is 3-colorable. If no,t then answer accordingly. If so, then add three new vertices, named R, B, and G,, connected in a triangle. We'll begin by coloring these vertices, as their names suggest, red, blue, and green. Next choose an uncolored vertex v. Add edges between v and two vertices from $\{R, B, G\}$. Ask the decision algorithm if the modified graph is 3-colorable. If it is, then color v the color of the unused vertex in $\{R, B, G\}$. If the graph is no longer 3-colorable, remove the added edges and repeat using a different pair from $\{R, G, B\}$. Repeat until all vertices are colored. (Note – the partial coloring is not input to the decision algorithm.)

3 Dimensional Matching: Let U, V, and W be disjoint sets such that $||U|| = ||V|| = ||W|| = q$. Let C be a subset of $U \times V \times W$; in other words C is a collection of triples (u, v, w), where $u \in U$, $v \in V$, $w \in W$. A 3 dimensional matching is a subset of C such that each element of U, V and W occurs in exactly one triple of C.

Input: Disjoint sets U, V, and W, where $||U|| = ||V|| = ||W|| = q$ and a subset C of $U \times V \times W$.

Question: Does C contain a three-dimensional matching?

Input: Disjoint sets U, V, and W, where $||U|| = ||V|| = ||W|| = q$ and a subset C of $U \times V \times W$.

Search Output: A three-dimensional matching from C, if one exists, and otherwise a statement that one does not exit.

Use the decision algorithm to determine if there is a three-dimensional matching. If not, answer accordingly. Otherwise, remove an unmarked triple from C. Ask the decision algorithm if the modified instance has a 3DM. If not, restore the triple and mark it as a member of the matching. Repeat until q triples are marked.

2. For each decision problem above do the following:

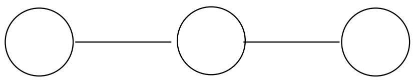
   (a) Give a yes-instance.

   (b) Give a no-instance.

   (c) Show the problem is in NP.

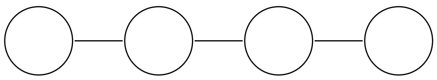
   Dominating Set:

   Yes instance:



K=1

No instance:

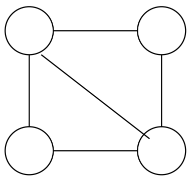


K=1

Given a set of vertices W that is supposed to be a dominating set of size at most K we can check that $||W|| \leq K$ and that every vertex in $V - W$ is adjacent to some vertex in W in polynomial time.

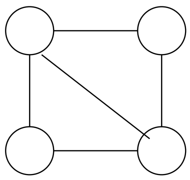
Clique:

Yes instance



K=3

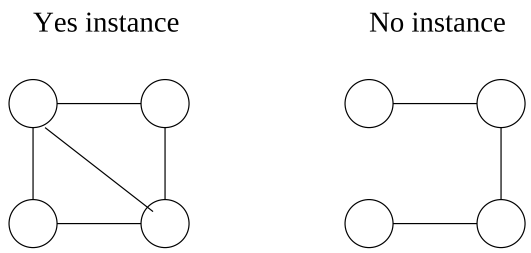
No instance



K=4

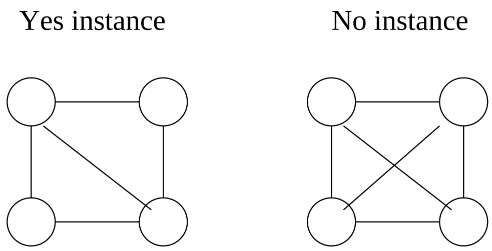
Given a set of vertices W that is supposed to be a clique of size at least K we can check that $||W|||geK$ and that every pair of vertices in W have an edge in G in polynomial time.

Hamiltonian Cycle:



Given a sequence of n vertices $v_1, v_2, \ldots, v_n, v_1$ that is supposed to be a Hamiltonian cycle we can check that each vertex occurs in the sequence and that v_i and v_{i+1} have an edge in G. K we can check that $||W|||geK$ and that every pair of vertices in W have an edge in G in polynomial time

3-Colorability:



Given a proposed coloring we can check that every vertex is uniquely colored and that no adjacent vertices have the same color in polynomial time.

3DM:

Yes instance: $U = \{u\}, V = \{v\}, W = \{w\}, C = (u, v, w)$

No instance: $U = \{u\}, V = \{v\}, W = \{w\}, C = \{\}$

Given a proposed matching we can check that each triple occurs in C and that each element of U, V and W occur in exactly one triple of the proposed matching.