CIS 675 (Fall 2018) Disclosure Sheet

Name: Rui Peng

HW #7

Did you consult with anyone on parts of this assignment, including other students, TAs, or the instructor?

Did you consult an outside source (such as an Internet forum or a book other than the course textbook) on parts of this assignment?

If you answered **Yes** to one or more questions, please give the details here:

By submitting this sheet through my Blackboard account, I assert that the information on this sheet is true.

This disclosure sheet was based on one originally designed by Profs. Royer and Older.

Homework 7:

Due December 5 at midnight Eastern time. Submit your solutions typed and in a pdf document. To receive full credit, explain your answers. In all of the problems below, assume $P \neq NP$.

If you collaborate with another student or use outside sources, please list those students' names and the URL/title/etc. of the sources that you referred to. Collaboration is permitted, but you must write up your own solutions.

Problem 1: The HITTING SET problem is as follows: Suppose we have a set of elements $V = \{v_1, ..., v_n\}$. Suppose we have a collection of sets $S_1, ..., S_k$, where each S_i is a set containing some of the v_j elements. A hitting set is a set H that is a subset of V, such that H contains at least one element from every S_i. Given some value b, we wish to find a hitting set with b or fewer elements, if it exists.

For example, suppose $S_1 = \{v_1, v_2, v_4\}, S_2 = \{v_2, v_3, v_5\}, S_3 = \{v_1, v_3, v_5\}$, and $b = 3$. Is there a hitting set with 3 or fewer elements? One example of a hitting set is $\{v_2, v_5\}$. This hitting set has size 2, so we have found a solution to the problem.

The VERTEX COVER problem is as follows: Given a graph G and some value c, is there a set of c vertices in the graph such that every edge in the graph is adjacent to at least one of those c vertices (in other words, every edge must have at least one of its two endpoints included in the set of c vertices)? (Note that this problem definition is slightly different from what we did in class.) VERTEX COVER is known to be NP-Complete.

Show that HITTING SET is NP-Complete by reducing the VERTEX COVER problem to HITTING SET.

Problem 2: Suppose we consider a variation of the SAT formula, called g-CLAUSE SAT, in which we are given a Boolean formula and an integer g. In the original SAT problem, we needed to assign truth values to literals such that every clause in the formula was satisfied. In this problem, we only need to satisfy g clauses. Show that g-CLAUSE SAT problem is NP-Complete by reducing SAT to g-CLAUSE SAT.

Problem 3: Suppose you are given an undirected graph G and a specified starting node s and ending node t. The HAMILTONIAN PATH problem asks whether G contains a path beginning at s and ending at t that touches every node exactly once. The HAMILTONIAN CYCLE problem asks whether G contains a cycle that touches every node exactly once (cycles don't have starting or ending points, so s and t are not used here).

Assume that HAMILTONIAN CYCLE is NP-Complete. Prove that HAMILTONIAN PATH is NP-Complete.

Problem 4: Describe reasonable heuristic algorithms or rules for the following problems. For this problem, you do not need to give full implementation details: it is enough to describe the main idea behind the algorithm or rule in 1-2 sentences. Keep in mind that it is ok if a heuristic sometimes fails to find the best answer.

1. You are designing an algorithm to control a robot that must navigate a maze and reach the center. The center is located at coordinates (0, 0), and the robot enters at coordinates (100, 100). You do not know the structure of the maze. What is a reasonable rule that the robot could follow to try to find the center?

2. PacMan is a popular video game in which you control a character in a maze. The maze contains golden tokens at various locations, and your goal is to 'eat' as many of these tokens as you can (in other words, move the PacMan character over the tokens). The maze also contains 'ghosts', who move around the maze and try to eat PacMan. If a ghost collides with PacMan, then the game is over. Design a reasonable heuristic algorithm for maximizing the number of tokens collected. (See https://www.google.com/#q=pacman&clb=clb to play.)

3. Consider a modified version of the SAT problem, in which the goal is to find a truth assignment for literals so that as many clauses as possible are satisfied. In other words, even if it is not possible to satisfy all clauses, your goal is to satisfy as many as you can. Design a reasonable heuristic algorithm for this problem.

4. The GRAPH COLORING problem is as follows: Given a graph G, we want to assign each node a 'color' such that if two nodes are adjacent (connected), then they do not have the same color. For example, if you are coloring countries on a map, then two countries that share a border should have different colors. The goal of this problem is to determine the smallest number of colors required to accomplish this. This problem is known to be NP-Complete. Design a reasonable heuristic algorithm for determining the number of colors needed to color the vertices so that no two adjacent vertices have the same color.

Extra Credit: In class, we covered the CLIQUE problem, in which we were given a graph G and had to find the largest clique in G. Recall that a clique is a set of nodes such that every node is connected to every other node in the set (i.e., everybody in the set knows everybody else in the set). The CLIQUE-3 problem is the same problem, except that we are *guaranteed* that every node has degree at most 3.

Suppose that I want to prove that CLIQUE-3 is NP-Complete. To do this, I make the following argument: We know that the CLIQUE problem is NP-Complete. CLIQUE-3 reduces to CLIQUE, because if we can solve CLIQUE for graphs in general, then clearly we can also solve it for graphs where the nodes have degree at most 3. Thus, CLIQUE-3 is also NP-Complete.

Part a: What is wrong with the above argument?

Part b: Show that CLIQUE-3 is *not* NP-Complete.