HW9

Due: Nov 8, 2018

Attach a fully filled-in cover sheet to the front of your printed homework. Your name should not appear anywhere; the cover sheet and each individual page of the homework should include your Banner ID only.

While collaboration is encouraged in this class, please remember not to take away notes from any labs or collaboration sessions.

Please monitor Piazza, as we will post clarifications of questions there. Hand in your solutions by 10:20am to the CSCI 1010 bin on the second floor of the CIT. Late homeworks are not accepted.

Problem 1

Ascii is a carpenter tasked with the immensely important task of constructing tiny digital tables so that internet users from all around the world can flip tables to their heart’s content. However, since he has to build so many tables, Ascii has created a machine that takes in stacks of tabletops and outputs completed tables. Now, he only has to build and stack the tabletops!

However, Ascii is running low on digital wood. In order to conserve materials, Ascii designs each tabletop to have holes that align with a two-column, $N$-row grid. So that the machine recognizes these stacks as valid stacks, Ascii needs to stack the tables such that for every position in the grid, at least one tabletop covers the position. (A tabletop covers a position in the grid if it does NOT have a hole in that position.)

Ascii is allowed to flip a tabletop along its vertical axis, but cannot rotate it or manipulate it in any other way, because otherwise the tabletops wouldn’t line up correctly. See Figure 1 for an example of a valid way to manipulate a tabletop.

Define TABLEFLIP to be the language consisting of sets of tabletops in which an arrangement exists such that every position in the grid is covered. Prove that TABLEFLIP is NP-complete. (Recall from Homework 8, Lab Problem 2 the things that need to be shown in order to prove that a language is NP-complete.)

**Hint:** Think of the orientation of each tabletop as an assignment to a
Problem 2

A classic NP-complete problem is determining whether a graph is $k$-colorable. Given a graph $G = (V, E)$ and a set of $k$ colors, is there an assignment of colors to vertices such that no two vertices that share an edge have the same color? A special case, 3-COLOR, is the problem in which there are 3 colors to choose from when making assignments. For this problem, you may assume that 3-COLOR is NP-complete.

$k$-COLOR $= \{ \langle G \rangle \mid G$ is a $k$-colorable graph $\}$

For all $k \geq 4$, show that $k$-COLOR is NP-complete.

**Hint:** First consider the case $k = 4$. 
The following questions are lab problems.

Lab Problem 1

AT&T is thinking of building more service towers in our area to give more customers dial-up internet access. You are charged with the task of choosing the best location to place these towers such that everyone is no more than $d$ distance away from a tower.

Show that this language is NP-complete:

\[ \text{AT&TInternetProblem}(\text{atip}) = \{ \langle G, d, k \rangle \mid \text{there is a set of } k \text{ vertices of graph } G \text{ at which towers can be positioned such that every vertex of } G \text{ is no more than } d \text{ edges from some tower} \} \]

Below is an instance of \text{atip} of a graph $G$, $k = 2$, and $d = 2$. We verify this by putting towers on vertices $v_3$ and $v_8$.

![Graph instance](image)

Lab Problem 2

Researchers at MySpace are trying to understand the social networks of their users by looking at networks of friend groups.

Define the language:

\[ \text{CliqueContest} = \{ \langle G, H \rangle \mid G, H \text{ are graphs, and there exists } k \text{ such that} \]
CSCI 1010 Due: Nov 8, 2018

G has a clique of size \( k \) but \( H \) does not.

a. Show that \textsc{CliqueContest} is NP-hard.

b. The class \textsc{coNP} is the class of languages whose complements are in \textsc{NP}. Show that \textsc{CliqueContest} is \textsc{coNP}-hard, i.e., that any language in \textsc{coNP} can be reduced to \textsc{CliqueContest} by a polynomial-time mapping reduction.

c. Show that if \( P = \textsc{NP} \), then there is a polynomial-time algorithm that decides \textsc{CliqueContest}.

d. Show that if \( \textsc{CliqueContest} \in \textsc{NP} \), then \( \textsc{NP} = \textsc{coNP} \).

Lab Problem 3

Altador is hosting its annual Altador Cup! This year, instead of the traditional Yooyuball, Neopets from across Neopia will be playing a much more fun game.

Consider the following single-player game, Hopscotch. It takes as input a sequence of positive integers \( [x_1, \ldots, x_n] \) and a target number \( t \). The numbers are written on the ground, and the player hops along the string of numbers—much like the classic playground hootenanny from which it derives its name. The player starts on the first number of the sequence. At each time step, they must jump to one of the next \( k \) numbers, where \( k \) is the number on which they are currently standing. The game ends when the player jumps to the last element of the sequence, at which point the player’s score is equal to the sum of the numbers they visited. The player wins if their score is equal to the target number.

An example is shown below with the sequence \( x = [1, 2, 5, 4, 10, 9, 6, 5, 1, 2] \) and target number \( t = 15 \). The current square is circled, past visited squares are italicized, and potential next squares are marked in bold. Note that this game progression represents one of several strategies which could be played from the same starting configuration.
In the second step, the player was forced to pick 2, but could then decide between 5 and 4. After picking 4, they decided between 10, 9, 6, and 5. Once they picked 6, there were fewer than six remaining numbers, so the player only had 5, 1, and 2 from which to choose. They picked 2, reaching the last element of the sequence. Their total score was $1 + 2 + 4 + 6 + 2 = 15$, reaching the target. Thus the player won this game.

Formally, we define $L$ to be the language of all winnable instances of Hopscotch:

$$L = \{ ([x_1, x_2, \ldots, x_n], t) \mid \text{there exists a sequence of valid Hopscotch moves such that the ending score is } t \}$$

Prove that $L$ is NP-complete.