Homework Assignment #6
Due: Dec. 9th, 2018

CMPUT 304
Department of Computing Science
University of Alberta

Note: All logarithms are in base 2 unless specified otherwise.
Exercises are for you and you alone. You need not submit them and they will not be graded.
You may submit as many answers to as many problems as you like.

Exercise I.
(i) Analogous to NP-completeness, define what is means for a language L to be coNP-complete. (Break down the two requirements to their definitions as well.)

(ii) Prove that the language Tautology is coNP-complete.

$$\textsc{Tautology} = \{\varphi,\ \text{a boolean formula}:\ \forall \bar{x}, \varphi(\bar{x}) = \texttt{true}\}$$

Exercise II. For each of the following languages determine whether it is in P or NP-complete, by either (briefly) describing a poly-time algorithm that decides it or by (briefly) describing a reduction from a NP-complete problem.

- Subgraph-Isomorphism: All pairs of graphs (G_1, G_2) such that there exists an injective mapping $\varphi : V(G_2) \to V(G_1)$ respecting the edge structure of G_2. Namely, a mapping φ from the vertices of G_2 into a subset of the vertices of G_1 such that $\forall u, v \in V(G_2)$ we have that (u, v) is an edge in G_2 iff $(\varphi(u), \varphi(v))$ is an edge in G_2.
- IntegerLinearProgramming (ILP): A feasibility LP with the additional constraints that all variables are integers.

Problem 1. (2 pts) Given a graph G, a 3-coloring of G is a function that maps the nodes of G into $\{R, G, B\}$ (red, green and blue). Given G and a 3-coloring $\chi : V(G) \to \{R, G, B\}$ we say an edge e is monochromatic if both endpoints of e are assigned the same color. We define the 3-Colorability language as

$$3\text{-}\textsc{Colorability} = \{G,\ \text{an undirected graph}:\ \exists \chi, \text{a 3-coloring of } G \text{ s.t. no edge is monochromatic}\}$$

Prove that 3-Colorability is NP-complete. (The reduction is described in CLRS, Question 34-3.)

Problem 2. (2 pts) We define the language

$$\textsc{Factoring} = \{(n, k),\ \text{two naturals}:\ \text{there exists a natural } 1 < x \le k \text{ s.t. } x \text{ divides } n\}$$

Prove that Factoring $\in$ NP $\cap$ coNP.

Problem 3. (2 pts) The SteinerTree problem is an extension of MST. In the SteinerTree problem you are given an undirected graph G and T, a set of *terminals* — a subset $T \subset V(G)$. Your goal is to find the smallest tree that spans (connects) all terminals. (So the MST problem is a special case where $T = V(G)$.) Correspondingly, we define the SteinerTree language:

$$\textsc{SteinerTree} = \{(G, T, k):\ \text{exists a Steiner Tree connecting all terminals in } T \text{ with } \le k \text{ edges}\}$$

Prove that VertexCover$\le_P$SteinerTree.

Hint: Think of each edge as a n-dimensional vector with exactly two 1-coordinates and all other coordinates set to 0. Now construct a graph where each node is associated with a n-dimensional vector, where one of the nodes is associated with the all-0 vector.

Problem 4. (6 pts) For each of the following languages determine whether it is in P or NP-complete, by either (briefly) describing a poly-time algorithm that decides it or by (briefly) describing a reduction from a NP-complete problem.
(i) (1 pts) <u>3-Colorability-1Blue</u>: All graphs where there exists a 3-coloring of the G such that no edge is monochromatic *and* exactly one node is colored 'Blue'.

(ii) (1 pts) <u>4-NAE</u>:
A 4-*not-all-equal* (NAE) clause C_{NAE} takes exactly 4 literals $C_{NAE}(l_1, l_2, l_3, l_4)$ as input, and is assigned the vale `true` if and only if not all 4 literals are identical (so if all 4 literals are `true` or all 4 are `false` then $C_{NAE}(l_1, l_2, l_3, l_4) = \texttt{false}$, otherwise $C_{NAE}(l_1, l_2, l_3, l_4) = \texttt{true}$). A 4-NAE boolean formula φ is the conjunction of m 4-NAE clauses. So now we define the language:

$$\text{4-NAE} = \{\varphi,\ \text{a 4-NAE boolean formula}: \quad \exists \bar{x} \text{ s.t. } \varphi(\bar{x}) = \texttt{true}\}$$

As a side note, observe that if $\bar{x}$ is an assignment under which $\varphi(\bar{x}) = \texttt{true}$ then negating all variables in $\bar{x}$ also results in an assignments that satisfies φ.

(iii) (1 pts) <u>3Out-Hamiltonian</u>: All directed graphs G for which there exists an almost Hamiltonian cycle — a simple cycle C that goes through all nodes in the graph except for at most 3 nodes.

(iv) (1 pts) <u>3Out-Cut</u>: All graphs such that there exists a way to partition the nodes into $(S, \bar{S})$ where all edges except for at most 3 cross the $(S, \bar{S})$-cut.

(v) (1 pts) <u>$(\Delta + 1)$-Colorability</u>: All graphs G where all degrees of all nodes are $\leq \Delta$, such that there exists a coloring $\chi : V(G) \to \{1, 2, 3, ..., \Delta, \Delta + 1\}$ (a coloring of the nodes of the graph with $\Delta + 1$ colors) under which no edge is monochromatic.

(vi) (1 pts) <u>Tripartiteness</u>: All graphs where one can partition the node set into 3 sets, S_1, S_2, S_3, such that any edge must cross either the cut $(S_1, S_2 \cup S_3)$ or the cut $(S_2, S_1 \cup S_3)$.