Exam Review

1. In class, we saw that 3-SAT is NP-complete. Consider the related problem 2-SAT: satisfiability of 2-CNFs. More formally, Given a CNF formula where each clause has exactly 2 literals/variables, output 1 if there is a satisfying assignment, and 0 otherwise. An example of a satisfiable 2-CNF is given below:

$$(\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3)$$

   We can express a 2-CNF using an implication graph $G = (V, E)$: For each variable/literal x_i in the 2-CNF, create vertices x_i and $\neg x_i$ in G. Then, for each clause $(x_i \vee x_j)$, create directed edges $(\neg x_i, x_j)$ and $(\neg x_j, x_i)$.

   (a) Draw what the implication graph would look like for the above 2-CNF example.

   (b) Using the notion of implication graph, describe a polynomial time algorithm to decide whether a given 2-CNF is satisfiable.

2. Consider the following two variants of the Independent Set problem:

   (i) Connected Independent Set: Given a graph $G = (V, E)$ and an integer k, determine if there exists an independent set of size k in G that is connected in the original graph.

   (ii) Independent Set on Trees: Given a tree T with n nodes, find the largest independent set, i.e., a set of nodes with no two nodes being adjacent.

   One of these problems is NP-complete, and one of these problems has an efficient algorithm. State which of them is NP complete, and use a reduction from the Independent Set problem to justify your answer.

3. For each of the two questions below, decide whether the answer is (i) Yes, (ii) No, or (iii) Unknown, because it would resolve the question of whether P = NP. Give a brief explanation of your answer.

   Let's define the decision version of the Interval Scheduling Problem from Chapter 4 as follows: Given a collection of intervals on a time-line, and a bound k, does the collection contain a subset of nonoverlapping intervals of size at least k?

   (a) Is it the case that Interval Scheduling $\leq_P$ Vertex Cover?

   (b) Is it the case that Independent Set $\leq_P$ Interval Scheduling?

4. Suppose it is proven that SAT$\in$P. I.e., there is some algorithm A which solves SAT in time $O(n^c)$ for some constant c.

   (a) We already have seen we can reduce SAT to/from any other NP-complete language. Does SAT$\in$P guarantee that SAT will reduce to any other problems in P? For instance, does SAT$\leq_p$MAX-FLOW? Which problems/languages does SAT poly-time reduce to if we assume SAT$\in$P?

   (b) Still assuming SAT$\in$P, suppose you want to actually find the satisfying assignment to a given formula $\phi(x)$, also in polynomial time. How could you use A to do so?

5. You are given a directed graph $G = (V, E)$ with weights we on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in G so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete.

6. State whether the following sets are countable, uncountable, or it depends.
   (a) $\mathbb{N}$
   (b) $\mathbb{Z}$
   (c) $\mathbb{Q}$
   (d) $\mathbb{R}$
   (e) Any subset of $\mathbb{R}$
   (f) Any open interval in $\mathbb{R}$
   (g) The set of all subsets of $\mathbb{N}$
   (h) The set of all strings over alphabet $\{0, 1\}$
   (i) The set of all finite strings over alphabet $\{0, 1\}$
   (j) The union of countable sets
   (k) The intersection of uncountable sets
   (l) The set of problems which are in P
   (m) The set of problems which are in NP
7. Hey wait a minute... P and NP are both countable, so there must be a bijection between problems in P and problems in NP. Why is this not a proof that P=NP?