CS2800 Design and Analysis of Algorithms
Problem Set #10 April 24, 2023

1. Consider the following problem called BoxDepth: Given a set of $n$ axis parallel rectangles (this means that the length and breadth of the rectangle are parallel to the $x$ and $y$-axes), how big is the largest subset of these rectangles that contain a common point.
   (a) Give a polynomial time reduction from BoxDepth to the problem of computing the largest clique in a graph.
   (b) Describe a polynomial-time algorithm to solve BoxDepth. Any polynomial-time algorithm would suffice. Don't try to optimize the running time.
   (c) Explain why this does not prove $\mathsf{P} = \mathsf{NP}$.

2. A formula is in disjunctive normal form (DNF) if it consists a disjunction of several terms, where each term is a conjunction of literals. For example $\phi = (x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge y \wedge \bar{z})$ is a DNF formula.
   (a) Give a polynomial-time algorithm to check if a DNF formula $\phi$ is satisfiable.
   (b) Consider the following "proof" of $\mathsf{P} = \mathsf{NP}$: Given a 3-CNF formula $\phi$, convert it to a DNF formula $\psi$ using the distributive law of $\wedge$ and $\vee$ as follows.

$$(x \vee y \vee \bar{z}) \wedge (\bar{x} \vee \bar{y}) = (x \wedge \bar{y}) \vee (y \wedge \bar{x}) \vee (\bar{z} \wedge \bar{x}) \vee (\bar{z} \wedge \bar{y})$$

   Now use the algorithm in Part (a) to check for satisfiability. What is wrong in this "proof"?

3. Given a graph $G(V, E)$, a $k$-*coloring* of the graph is an assignment of colors $c : V \longrightarrow \{1, 2, \ldots, k\}$ such that if $(u, v) \in E$, then $c(u) \neq c(v)$. The 3-Colorability problem is to test if a given graph has a 3-coloring. This problem is known to be $\mathsf{NP}$-complete.

   Suppose that you are given a subroutine, that takes as input a graph $G$, and decides whether $G$ is 3-colorable or not. Using this subroutine, and assuming that a call to the subroutine takes constant time, design a polynomial time algorithm to compute a 3-coloring $c : V \longrightarrow \{1, 2, 3\}$, if one exists.

4. A Hamiltonian path in an undirected graph $G(V, E)$ is a simple path in $G$ that passes through each vertex in $G$ exactly once. Similarly, a Hamiltonian cycle is a cycle that starts from a vertex $v \in V$, passes through every other vertex exactly once and returns to $v$. The HamPath and HamCycle problem is to decide if a given input graph $G$ has a Hamitonian path and Hamiltonian cycle respectively.
   (a) Show that HamPath $\leq$ HamCycle.
   (b) Show that HamCycle $\leq$ HamPath.

5. In this question, we will look at some variants of Sat and their complexity.

(a) Let $\phi$ be a CNF formula such that each clause contains exactly 3 literals, and each variable occurs in at most 3 clauses. Show that $\phi$ is satisfiable.

To do this, the following statement (known as Hall's theorem) may be useful.

> A bipartite graph $G(L \cup R, E)$ such that $|L| < |R|$ has a matching $M$ that matches every vertex in $L$ if and only if for every $L' \subseteq L$, $|L'| \leq |N(L')|$, where the $N(L')$ is set of neighbors of $L'$.

(b) Let $\phi$ be a CNF formula where each clause has at most 3 literals, and each variable appears in at most 3 clauses. Show that this variant of SAT is NP-complete.

**Hint:** Replace each occurrence of a variable with a new copy. Add constraints to say that all the copies must take the same truth value.

(c) Suppose that $\phi$ is a CNF formula where each clause has exactly 3 literals, and each variable occurs in at most 4 clauses. Show that this variant of SAT is NP-complete.

**Hint:** For each clause $C_i$ with two literals, add a new variable $x_i'$ into it. Put additional clauses (with different variables) to force $x_i'$ to take the truth value 0.

(d) Given a CNF formula $\phi$ such that each clause contains an arbitrary number of literals, but each variable occurs in at most 2 clauses, give a polynomial-time algorithm to check if $\phi$ is satisfiable.

**Hint:** Sometimes greed is good.