CS 580 Final Practice Problems

Note that for the questions stating: "Either give an efficient (polynomial time) algorithm for this problem or prove that it is NP-Hard." You have to choose *one* out of these two options:

- Give an efficient polynomial time algorithm. Solutions with suboptimal time complexity will only be awarded partial points. If you propose an algorithm in any problem, then you also need to prove correctness.
- Prove that the stated problem is NP-Hard.

If you propose an algorithm in any problem, then you also need to prove correctness.

1. Prove that every undirected graph G with maximum degree Δ has a valid vertex coloring of $\Delta + 1$ colors.
2. The k-COLOR problem is as follows: Given an undirected graph G, decided whether there exists a valid vertex coloring using k colors. Show that 4-COLOR reduces to solving 3-COLOR.
3. Suppose that as a birthday present you are given an oracle O that solves the decision problem for Independent Set — given a graph G and a number $k > 0$ the oracle O(G,k) outputs YES if G contains an independent set of size k and NO otherwise. Recall that a subset $S \subseteq V(G)$ is independent if for every pair of nodes $u, v \in S$ with the graph G does not contain the edge $\{u, v\}$. Develop an polynomial time oracle algorithm $\mathcal{A}(G)$ to find the largest independent set in G. Your algorithm should make polynomially many queries to the oracle O.
4. In the CYCLE-REMOVAL problem, you are given an undirected simple graph $G = (V, E)$ and an integer k, and the goal is to decide if there exists a set $S \subseteq V$ with $|S| \leq k$, upon whose removal the remaining graph contains no cycles. In other words, removing S and the edges adjacent to vertices in S results in a graph with no cycles. Either give an efficient (polynomial time) algorithm for this problem or prove that it is NP-Hard.
5. A boolean formula $\Phi = C_1 \wedge \ldots \wedge C_m$ is a 4CNF formula if each clause contains exactly 4 variables. The decision problem $\mathsf{4SAT}$ is defined as follows: given a 4CNF formula Φ, decide whether or not Φ is satisfiable. Either give an efficient (polynomial time) algorithm for this problem or prove that it is NP-Hard.

6. Suppose we have a complete graph on n vertices. Show (for small enough values of n) that it is possible to color the edges of the graph in two colors (say red and blue) so that there is no complete subgraph on r vertices which is monochromatic (every edge colored the same color).

7. Given an undirected graph G. Does there exists a vertex cover of size atmost 20? Either give an efficient (polynomial time) algorithm for this problem or prove that it is NP-Hard.

8. Suppose that you have m balls and one bin. For each ball you throw it in the bin independently with probability p. Let X be the number of resulting balls in the bin. What is the expected value of X? Give an upper bound on the probability $\Pr[X \geq 2E[X]]$.

9. Write the dual to the following linear program.

$$\begin{aligned} \max\ & 2x + 3y \\ & 2x + y \leq 10 \\ & x + y \leq 8 \\ & x + 3y \geq 2 \\ & x, y \geq 0 \end{aligned}$$

   Draw the feasible region of the above LP.

10. Let MAX EXACT3SAT be the problem where each clause has exactly 3 literals and the goal is to find an assignment that maximizes the number of clauses satisfied. We know that the following holds: MAX EXACT3SAT does not admit a α-approximation algorithm for any $\alpha > 7/8$, unless $P = NP$.

    Consider the problem MAX INDEPENDENT SET defined as follows. The input is an undirected graph and the goal is to find an independent set of maximum cardinality. Prove that the MAX INDEPENDENT SET problem also does not admit a α-approximation for any $\alpha > 7/8$ unless $P = NP$.

11. Consider the *vertex cover* problem. The input is a graph G and a non-negative weight function w over its vertex set. A valid output is any subset of vertices that covers all edges. The goal is to minimize the sum of weights of vertices in the output set i.e. find a minimum weight vertex cover.

    We say that an edge $e : (u, v)$ is not covered by a set $S \subseteq V$ if $u, v \notin S$. Consider the following algorithm:

    - While there exists an edge $e : (u, v)$ such that $\min\{w(u), w(v)\} > 0$:
      - Let $\alpha = \min\{w(u), w(v)\}$.
      - Subtract α from each $w(u)$ and $w(v)$ i.e. set $w(u)$ to $w(u) - \alpha$ and $w(v)$ to $w(v) - \alpha$.

- Return $\{v : w(v) = 0\}$.

Prove that this is a 2-approximation algorithm.