Exercises for Submission

1. A *maximum independent set* is a largest independent set in a given graph. A *maximal independent set* is an independent set such that adding any other vertex to the set forces the set to contain an edge. Assuming P $\neq$ NP, is there a polynomial-time algorithm that finds a maximum independent set in a given graph G? And what about maximal independent set?

2. Let 10TSP be a variant of TSP, in which the weight function w is symmetric (i.e., $w(x, y) = w(y, x)$ for every pair of vertices $x, y \in V$) and instead of triangle inequality, for every three vertices $x, y, z \in V$ we have $w(x, y) \leq w(x, z) + 10 \cdot w(z, y)$. Suppose we apply the 2-approximation algorithm for TSP-with-triangle-inequality from class on 10TSP instances. What is the approximation ratio we get? Try to make your analysis tight.

3. (a) Consider the greedy algorithm for the Vertex Cover problem: Choose a vertex of maximal degree. Add it to the cover, and remove it and all its edges from the graph. Repeat until no edges remain in the graph.
   Prove that this algorithm is *not* a 2-approximation.

   (b) Show a polynomial-time algorithm that given a graph $G = (V, E)$ which contains an independent set of size $\frac{3}{4}|V|$ finds an independent set of size at least $\frac{1}{2}|V|$.

4. The problem Maximum Coverage is defined as follows:
   Input: A universe U of n elements, where the ith element has some non-negative weight w_i, a family of t subsets $S_1, \ldots, S_t \subseteq U$, and a number k.
   Output: A set of k indices $\{i_1, \ldots, i_k\}$ such that $\bigcup_{j=1}^{k} S_{i_j}$ has maximal weight.
   Describe the greedy algorithm for the Maximum Coverage problem, and prove that it achieves an approximation factor of at least $1 - e^{-1}$.

5. (a) Prove that graphs with maximum degree Δ are $(\Delta + 1)$-colorable, and show a polynomial-time algorithm that finds a $(\Delta + 1)$-coloring.

   (b) Prove that if there exists a 2-approximation polynomial-time algorithm for the maximum independent set problem, then there exists a polynomial-time algorithm that colors a given 3-colorable graph using $O(\log n)$ colors, where n denotes the number of the vertices.

   (c) Show a polynomial-time algorithm that finds an $O(\sqrt{n})$-coloring of a 3-colorable graph, where n denotes the number of the vertices. Hint: Use (5a) and the fact that we can find a 2-coloring of a 2-colorable graph in polynomial-time.

6. We showed in class a reduction from NAE-3SAT to MAX-CUT on multi-graphs. Use this reduction to show that there exists a constant $c < 1$ such that MAX-CUT on multi-graphs cannot be approximated to within c unless P = NP.