Computational Complexity Practice Questions

1. Show that any array of integers $x[1 \ldots n]$ can be sorted in $O(n+M)$ time, where $M = \max_i x_i \min_i x_i$. Write succinct pseudocode for your method. For small M , this is linear time: why doesn't the $\Omega(nlogn)$ lower bound apply in this case?

2. Show that the following problems are in NP (and give the poly-time verifiers):

   COMPOSITE Is the integer x the product of two other integers?

   SET COVER Given a collection of sets $S_1, \ldots, S_m$ of a ground set and an integer k, determine if there exists a subcollection of size k such that the union of the sets in the subcollection is the same as the union of the sets in the collection.

   SAT (See the readings for a definition.)

   ROOT Given a polynomial $P(x)$ (e.g. $P(x) = x^3 - 3x + 1$) and an interval $[a, b]$, does P have a root in $[a, b]$, i.e. does there exist an integer x s.t. $a \leq x \leq b$ s.t. $P(x) = 0$ (What if x is not required to be an integer?)

3. We are feeling experimental and want to create a new dish. There are various ingredients we can choose from and we'd like to use as many of them as possible, but some ingredients don't go well with others. If there are n possible ingredients (numbered 1 to n), we write down an $n \times n$ binary matrix where the (i, j) entry is 1 if i and j can go together and 0 otherwise. Notice that this matrix is necessarily symmetric; and that the diagonal entries are always 0.

   We wish to solve the following problem:

   EXPERIMENTAL CUISINE :

   input: n, the number of ingredients to choose from; B, the $n \times n$ binary matrix that encodes which items go well together

   output: the maximum number of ingredients which can be selected together

   Show that if EXPERIMENTAL CUISINE can be solved in polynomial time, then P=NP.

4. Prove NP-completeness by generalization: For each of the problems below, prove that it is NP- complete by showing that it is a generalization of some NP-complete problem in the readings.

   (a) SUBGRAPH ISOMORPHISM: Given as input two undirected graphs G and H, determine whether G is a subgraph of H (that is, whether by deleting certain vertices and edges of H we obtain a graph that is, up to renaming of vertices, identical to G), and if so, return the corresponding mapping of $V(G)$ into $V(H)$.

   (b) LONGEST PATH: Given a graph G and an integer g, find in G a simple path of length g.

   (c) SET COVER