Due Date: October 21, 2014

Remark: Prove the correctness of every algorithm and analyze its running time.

Problem 1: Given a set $X = \{X_1, X_2, \ldots, X_n\} \subset \mathbb{N}$ and a number $t \in \mathbb{N}$, give an $O(nt)$ time algorithm that either returns a subset of $X$ which sums to $t$ if there exists one, or reports that such a subset does not exist.

Problem 2: Prove that if $\text{co-NP} \neq \text{NP}$ then $\text{NP} \neq P$.

Problem 3: Given an undirected graph $G = (V, E)$ and a positive integer $k$, show that it is $\text{NP}$-complete to decide whether there exists a subset $X \subseteq V$ with $|X| \leq k$ such that deleting $X$ from $G$ makes it acyclic.

Problem 4: Suppose we are given a set $X = \{x_1, \ldots, x_q\}$ and a collection $C = \{C_1, \ldots, C_k\}$ of 3-element subsets of $X$ ($q \leq k$). Show that it is $\text{NP}$-complete to decide whether $C$ contains an exact cover for $X$, i.e., a subset $C' \subseteq C$ such that every element of $X$ occurs in exactly one member of $C'$.

Problem 5: Given disjoint sets $X$, $Y$, and $Z$, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, a subset $M \subseteq T$ is a 3-dimensional matching if each element of $X \cup Y \cup Z$ is contained in at most one of these triples. The maximum 3-dimensional matching problem is to find a 3-dimensional matching $M$ of maximum size. Give a polynomial-time algorithm that finds a 3-dimensional matching of size at least $1/3$ times the maximum possible size.