1. Given a directed graph $G$=($V$, $E$), and two vertices $u$ and $v$ in $V$, we call vertex $v$ is ***reachable*** from $u$, if there exists a directed path from $u$ to $v$. A vertex $s$ in $V$ is called a ***source vertex*** if every vertex in $V$ is reachable from $s$.
   a) Given a directed graph $G$=($V$, $E$), and a specified vertex $v$, design a linear time algorithm (i.e. your algorithm should run in $\Theta(|V|+|E|)$ time) to determine if $v$ is a source vertex. You need to describe the data structure used in your algorithm. (10%)
   b) Given a directed acyclic graph (DAG; a directed graph is acyclic if it contains no directed cycles) $G$=($V$, $E$), you are asked to determine if $G$ contains a source vertex. If you apply the algorithm of subproblem (a) on every vertex of $G$, you will get an algorithm runs in $\Theta(|V|^2+|V||E|)$ time. It is not desirable. Design a more efficient algorithm for this problem. Analyze the time complexity of your algorithm. (10%)

2. Design an algorithm to determine whether a given undirected graph with $n$ vertices contains a cycle of length 4 as a subgraph. There is a naïve $\Theta(n^4)$ time algorithm that solves this problem by checking all possible 4 vertices of the graph. Your algorithm must be more efficient than this algorithm. (20%)

3. For a set of variables $x_1, x_2, \ldots, x_n$, you are given some equality constraints, of the form "$x_i = x_j$" and some disequality constraints, of the form "$x_i \neq x_j$". Is it possible to satisfy all of them? For instance, the constraints :
   $x_1 = x_2$,; $x_2 = x_3$; $x_3 = x_4$; $x_1 \neq x_4$;
   cannot be satisfied. Give an efficient algorithm that takes as input $m$ constraints over $n$ variables and decides whether the constraints can be satisfied. Describe the data structure used by your algorithm, and analysis the time complexity of your algorithm. (20%)

4. Consider the following decision problem called "Square" appeared in the problem set of a program contest: Given a set of sticks of various lengths, is it possible to join them end-to-end to form a square? For example, if the given lengths are 10, 20, 30, 40, 50, then the answer is "no"; if they are 1, 7, 2, 6, 4, 4, 3, 5, then the answer is "yes".
   a) Show that this problem is an NP problem. (10%)
   b) Prove that it is NP-hard. (10%)
   c) If the total sum of lengths is a polynomial of $n$, the number of given lengths, then it is possible to solve this problem in polynomial time. Design such an algorithm to solve this problem. (20%)