CS 213
Homework Assignment 3

Given: February 05, 2009 **Due:** February 12, 2009

This assignment is due by the end of the class on the due date. Unless all problems carry equal weight, the point value of each problem is shown in [ ]. To receive full credit all your answers should be carefully justified. Each solution must be the student's own work. Assistance should be sought or accepted only from the course staff. Any violation of this rule will be dealt with harshly.

---

1. Given a sorted array $A[1..n]$ of reals, give an efficient algorithm to determine whether A has a *majority* element or not. A majority element is an element that appears more than $n/2$ times.

2. Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. (It should not output all cycles in the graph, just one of them). The running time of your algorithm should be $O(m+n)$ for a graph with n nodes and m edges.

3. We have a connected graph $G = (V, E)$, and a specific vertex $u \in V$. Suppose we compute a depth-first search tree rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute breadth-first search tree rooted at u, and obtain the same tree T. Prove that $G = T$. (In other words, if T is both a depth-first search tree and a breadth-first search tree rooted at u, then G cannot contain any edges that do not belong to T.

4. Given an undirected graph $G = (V, E)$ and an integer k, find an induced subgraph $H = (U, F)$ of G of maximum size (maximum in terms of the number of vertices) such that all vertices of H have degree at least k, i.e., each vertex in H has at least k neighbors in H.