CS41 Fall 2012 Homework 2: Due in class Thursday September 27th.
You may work with one partner on this assignment. If you work together, you only need to submit one set of written solutions.

1. (2.5) Let f and g be two functions such that $f(n)$ is $O(g(n))$. For each of the following statements, decide if the statement is true or false. If it is false, give a counter example. If it is true, give a proof.

   (a) $\log_2 f(n)$ is $O(\log_2 g(n))$.
   (b) $2^{f(n)}$ is $O(2^{g(n)})$.
   (c) $f(n)^2$ is $O(g(n)^2)$.

   Finally, let p and q be two functions such that $\log_2 p(n)$ is $O(\log_2 q(n))$. Is $p(n)$ $O(q(n))$? Give a proof if true, or a counterexample if false.

2. Let G be a DAG with two vertices s and t such that s has in degree zero and t has out degree zero. Design an algorithm to count the number of distinct paths from s to t in G. Paths p and p' are distinct if they differ by at least one vertex along the paths. Thus $p = \langle s, v, w, t\rangle$, $p' = \langle s, v, z, t\rangle$ and $p'' = \langle s, v, t\rangle$ are all distinct. Analyze the run time of your algorithm and give a proof of its correctness.

3. (3.9) Let G be an undirected graph with n vertices consisting of a single connected component. Suppose there are two vertices s and t such that the shortest distance from s to t is strictly greater than $n/2$. Show that there exists a vertex $v \neq s \neq t$, such that the removal of v and its adjacent edges from G results in a disconnected graph with no path from s to t. Design an algorithm to find such a vertex, and analyze its run time.