Solution of Assignment #1 part 2, Total points: 45
(Course: CS 401)

Problem 1 (30 points): Assume that you have two functions $f(n)$ and $g(n)$ such that $f(n) = O(g(n))$. Also, assume that $f(n) \geq 4$ and $g(n) \geq 4$ for all n. For each of the following statements, decide whether you think it is true or false and accordingly give a proof (if true) or a counter-example (if false).

***(i)* (10 points)** $\log_2 f(n)$ is $O(\log_2 g(n))$.

***(ii)* (10 points)** $2^{f(n)}$ is $O\left(2^{g(n)}\right)$.

***(iii)* (10 points)** $f(n)^2$ is $O\left(g(n)^2\right)$.

Solution:

(i) **True**.

$$f(n) = O(g(n)) \Rightarrow f(n) \leq c_1 g(n) \Rightarrow \log_2 f(n) \leq \log_2 c_1 g(n) = \log_2 c_1 + \log_2 g(n)$$

$$= \left(\frac{\log_2 c_1}{\log_2 g(n)} + 1\right) \log_2 g(n) \leq (\log_2 c_1 + 1) \log_2 g(n) \leq c_2 \log_2 g(n)$$

Here we can choose any constant c_2 as long as $c_2 \geq \log_2 c_1 + 1$.

(ii) **False**. e.g. $f(n) = 2n$ and $g(n) = n$.
$2n = O(n)$, but $2^{2n} = 2^n \times 2^n$. We cannot find a constant c, such that $2^n \times 2^n \leq c2^n$ holds for any $n \geq n_0$

(iii) **True**.

$$f(n) = O(g(n)) \Rightarrow f(n) \leq cg(n) \Rightarrow f(n)^2 \leq (cg(n))^2 = c^2 g(n)^2$$

Problem 2 (15 points): Give an algorithm to detect whether a given undirected graph is a tree or not. The graph is given to you in its adjacency list representation. The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.

Solution: Let G be the given graph. We run BFS starting from an arbitrary node s. If BFS cannot reach all nodes then the graph is not connected and hence not a tree. Otherwise, consider the obtained BFS tree T. If every edge of G appears in the BFS tree then $G = T$, hence G contains no cycle and therefore G is a tree. Otherwise, G is a non a tree by the following argument. There is some edge $e = \{v, w\}$ that belongs to G but not to T and thus G has strictly more than $n - 1$ edges.