CS 3343 Assignment 4 (corrected)
Due Thursday, October 31, 1996

The problems are not equally weighted.

1) Read problem 7.1 on page 250 of the text. Consider the following representation of an adjacency list. The nodes are numbered 0 through N-1, where N is a constant. Assume that the graph is connected.

```
typedef struct list_entry {
   int node;
   struct list_entry *next;
} list_entry_t;

list_entry_t adjlist[N];
```

Assume that the next entry is -1 for the last entry in the list. Write the function:
`void addarc(int n, int m, list_entry_t adj[]);`
which inserts an arc between nodes n and m. Assume that the arc is not already in the graph.
Write the function: `int delarc(int n, int m, list_entry_t adj[]);`
which deletes the arc between nodes n and m in the graph and return 0 if this is possible and returns 1 if it is not possible. You can assume that the adjacency list is valid and that the graph is still connected after the arc has been deleted.

2) Problem 7.3 on page 251 of the text.

3) For what values of i and j can $A(i,j)$ (as discussed in class) by represented by a 64-bit unsigned integer?

4) Order the following 12 functions from smallest order to largest order. Each function will be $f_i(n)$ for some i between 1 and 12, such that $f_1(n) = O(f_2(n))$, $f_2(n) = O(f_3(n)) \ldots f_{11}(n) = O(f_{12}(n))$. For each value of i between 1 and 11 explain why $f_i(n) = O(f_{i+1}(n))$. You may use the definition of O(f(n)) or l'Hôpital's rule, or another method. For which values if i is $f_{i+1}(n) = O(f_i(n))$?

$\lg(\lg^* n)$	$\lg^*(\lg n)$	$\lg(n!)$
$\lg((n+1)!)$	3^n	$(\lg n)^{2^n}$
4^n	$\left(\frac{1}{2}\right)^n$	n
$2^{\lg n}$	$2^{\lg^* n}$	$n^{\lg \lg n}$

5) For each of the following determine if it is true or false. It is true if it holds for all functions f and g which are always greater than 2. If true, show why. If false, give a counterexample.
   a) if $f(n) = O(g(n))$ then $f(n)^2 = O(g(n)^2)$
   b) if $f(n) = O(g(n))$ then $\lg(f(n)) = O(\lg(g(n)))$
   c) if $f(n) = O(g(n))$ then $2^{f(n)} = O(2^{g(n)})$