CS715 Midterm Examination, March 25, 2008. No books or notes.

1. Find the solutions to the following recurrences. Prove each answer. You may use the following three statements without proof:

   - Let f' be the derivative of a function f.
     If $F(n) = F(n-1) + f'(n)$, then $F(n) = \Theta(f(n)$, for most f that you would be interested in.
   - If $F(n) = F(\log n) + 1$ then $F(n) = \Theta(\log^* n)$.
   - The Master Theorem.

   (a) $G(n) = 4G\left(\frac{n}{2}\right) + n^2$

   (b) $F(n) \leq 2F(\sqrt{n}) + \log n$

   (c) $H(n) = 2H(n-1) + 1$

2. Walk through the steps of the REDUCE procedure of the SMAWK algorithm for the following strictly monotone matrix:

   DURING THE ACTUAL TEST, THERE WILL BE A MATRIX. MEANWHILE, TRY YOUR HAND ONE OF THE FIVE PRACTICE MATRICES.

3. Briefly explain how Johnson's algorithm works. Do not write pseudo-code.

4. Write pseudo-code for the Floyd-Warshall algorithm. Be sure to include a branch that indicates that a negative cycle has been detected. If it takes more than half a page, you are writing too much.

5. - Given a Boolean expression $E(x_1, x_2, \ldots x_k)$, where each x_i is a free variable of Boolean type, we say that E is *satisfiable* if there exists a Boolean vector $a = (a_1, a_2, \ldots a_k)$ such that $E(a)$ is true.

     The Boolean satisfiability problem is defined as follows. Given a Boolean expression E, is E satisfiable? Karp proved that the Boolean satisfiability problem is $\mathcal{NP}$-complete.
   - *3-satisfiability* is a modified version of the Boolean satisfiability problem. Given a Boolean expression E in *conjunctive normal form*, where each clause has three terms, is E satisfiable? 3-satisfiability is also $\mathcal{NP}$-complete. (We proved that in CS 456, by reducing Boolean satisfiability to 3-satisfiability.) For more details, consult the *world-wide web* or an appropriate book.
   - Given a graph $G = (V, E)$, a *vertex cover* of G is a set of vertices $S \subseteq V$ such that every $e \in E$ meets some $v \in S$.

     The $0-1$ *vertex cover problem* (also called the *decision* version of the problem) is defined as follows. Given a graph $G = (V, E)$ and an integer k, does there exist a vertiex cover of G of cardinality k?

   Prove that the $0-1$ vertex cover problem is $\mathcal{NP}$-complete by reduction, using the fact that 3-satisfiability is $\mathcal{NP}$-complete.

6. - Assume that george(S, x) is a function that returns a Boolean value, where S is a stack, and that the time complexity of george is $O(\log |S|)$, where $|S|$ is the current size of the the stack.
   - Consider the following block of pseudocode.

```
S ← the empty stack
for i from 1 to n do
    read(x_i)
    while (S ≠ ∅) and george(S, x_i) do
        pop(S)
    end while
```

push x_i onto S
end for

- Assume that reading, popping, and pushing take $O(1)$ time each.

Use amortized analysis to prove that the block of code runs in $O(n \log n)$ time.