Computer Science 715 Spring 2010 Final Examination, May 1, 2010

1. (a) Consider the algorithm EIGByz (given in class, and in Lynch) for the binary Byzantine Generals problem, where $n = 4$ and $f = 1$. Suppose we select the default bit $v_0 = 0$. Describe a situation where, after all messages of EIGByz are sent and received, every loyal general has enough information to prove that there is no common input, and yet all of them decide on the answer 1.

   (b) We are given that EIGByz is correct, provided $n > 3f$. Describe an algorithm for the binary Byzantine Generals problem for $n > 3f$, which has the following property:

   > If all loyal generals can prove, based on their input and messages received so far, that there is no common input, then all loyal generals will select the default bit.

2. (Complexity, *etc.*)

   (a) The *degree* of a vertex of a graph is the number of neighbors of that vertex. The degree of a graph is the maximum degree of any vertex of the graph. Suppose that G is a connected graph with n vertices, whose degree and diameter are both d. Prove that $d = \Omega\left(\frac{\log n}{\log \log n}\right)$.

   (b) Suppose α and β are positive real numbers, and that $\alpha + \beta < 1$. Solve the recurrence:

   $$F(n) \leq F(\alpha n) + F(\beta n) + n$$

   (c) Solve the recurrence:

   $$F(n) = \sqrt{n}F(\sqrt{n}) + n$$

   (d) Solve the recurrence:

   $$F(n) = \frac{n}{\log n}F(\log n) + n$$

   (e) Use amortized analysis to prove that the following pseudo-code takes $O(n)$ time to execute. The function UNKNOWN has Boolean type, and returns whatever the adversary decides each time it is called.

   ```
   m = 0
   for i = 1 to n do
       while m > 0 and UNKNOWN do
           m = m − 1
       end while
       m = m + 1
   end for
   ```

   (f) Using the result from Problem 2e, prove that the REDUCE subroutine of SMAWK takes $O(n + m)$ time to reduce the case of an $n \times m$ strictly monotone matrix to the case of an $n \times n$ strictly monotone matrix.

3. Give a solution to the range query problem of size n that takes $O(n \log^* n)$ preprocessing time and $O(1)$ time for each query, and which uses $O(n \log^* n)$ space.

4. (a) Here is the usual definition of $O(g(n))$.

   "Suppose that f and g are functions defined on positive integers. We say that $f(n) = O(g(n))$ if there exist a constant C and an integer N such that $f(n) \leq Cg(n)$ for all $n \geq N$."

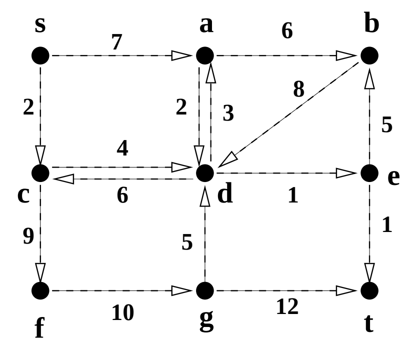
   When we say that the error of a Runge-Kutta method is $O(h^5)$ (or whatever) the above definition clearly does not apply. Give a definition that works for that case.

   (b) Suppose we are given an ordinary differential equation $y' = f(x, y)$ and the boundary condition that $y = 0$ when $x = 0$. Assume that f is analytic for all pairs (x, y) in the square $0 \leq x \leq 1$, $0 \leq y \leq 1$, and that $0 \leq f(x, y) \leq 1$ for all pairs (x, y) in that square.

   You wish to compute the value of y when $x = 1$. Prove that if you use the third order Runge-Kutta method we discussed in class, using steps of size h, where h is a small positive number, you will compute the value with an error of $O(h^3)$.

5. Use the Fast Fourier Transform to multiply 75 by 49. Show all steps.

6. Find the max-flow and the min-cut of the weighted directed graph pictured below.



7. Given a sequence of integers, it is possible to find the longest monotone increasing subsequence in $O(n \log n)$ time, where n is the length of the sequence. Explain how that is done.

   Walk through your algorithm for the example sequence

$$6, 8, 3, 12, 10, 2, 4, 11, 1$$

   At each step, illustrate the current state of the data structure.

8. Explain Johnson's algorithm for the all-pairs shortest path problem on a sparse weighted directed graph of n nodes and m edges where there are no negative cycles (although there could be negative edges). Give the asymptotic time complexity in terms of n and m.

9. A triangular matrix $\{M[i, j]\}_{0 \leq j < i \leq n}$ is said to be *reverse Monge* if $M[b, c] + M[a, d] \leq M[c, a] + M[d, b]$ for all $0 \leq c < d < a < b \leq n$.

   Can you give an efficient algorithm for finding all row minima of a reverse Monge matrix? Be sure to give the time complexity.