If you don't like exams, think of this as a solo homework. Unlike the previous homeworks, no collaboration is allowed; don't talk/email/text/post/sing/mime about the exam with each other or with anyone else (except Jeff) until after the due date. ("The first rule of Fight Club is: There is no Fight Club. The second rule of Fight Club is: *There is no Fight Club.*") You are welcome to consult anything written, either on paper or electronically, before May 2003. As usual, please cite your sources.

If you discover a bug in one of the questions, send me email—I'll post updates to the course newsgroup. For extra credit, answer the question I *intended* to ask!

1. Recall that an *n-bit binary-to-positional converter* B_n is a circuit with n input bits $x_1, \ldots, x_n$, representing an integer x in binary, and 2^n output bits $y_0, y_1, \ldots y_{2^n-1}$, where $y_i = 1$ if and only if $i = x$. Show that B_n can be realized by a circuit consisting of $O(2^n)$ binary AND and OR gates and no NOT gates.

2. The *st-connectivity problem* is defined as follows: Given the adjacency matrix A of an undirected graph G, decide whether there is a path from the first vertex to the last vertex in G (implicitly represented by the first and last rows of A). Here, s and t stand for 'source' and 'target'.

   (a) Prove that st-connectivity is evasive, that is, any boolean decision tree that determines whether s and t are connected must examine every bit in the adjacency matrix.

   (b) Prove that the st-connectivity problem can be solved using a circuit of depth $O(\log^2 n)$, using only standard gates. *[Hint: Use a communication protocol.]*

   (c) How many gates does your circuit have?

3. A *perfect matching* is an undirected graph where every vertex has degree exactly 1, or equivalently, a graph with n vertices and $n/2$ edges, where no two edges share a vertex.

   (a) Prove that being a perfect matching is an evasive graph property.

   (b) Prove that the *randomized* decision tree complexity of determining whether a graph is a perfect matching is $\Omega(n^2)$. *[Hint: Use Yao's Lemma.]*

4. Given a list of n values $x_1, x_2, \ldots, x_n$, the *majority element problem* is to determine the value (if any) that occurs more than $n/2$ times in the list. This problem asks you to bound the complexity of this problem, in a decision tree model where every decision is an equality test of the form "$x_i = x_j$?"

   (a) Prove that the deterministic complexity of the majority element problem is exactly $\lfloor 3(n-1)/2 \rfloor$.

   (b) Find the bext upper and lower bounds you can for the *randomized* complexity of the majority element problem.

*5. Recall that the property of having three consecutive 1s is an evasive string property when $n \bmod 4 = 0$ or 3. Prove the best upper and lower bounds you can on the *randomized* decision tree complexity of this string property. *[Hint: The best bounds I know of are $\geq 9n/16$ (using Yao's Lemma) and $\leq 7n/8$.]*