Problem of the week #5:
Given - Feb. 13, 2003; Due - Feb. 20, 2003 (in class).

You are required to analyze the following routing algorithm to route an arbitrary permutation on a $\sqrt{N} \times \sqrt{N}$ 2-dimensional array of processors. Assume that edges are bi-directional i.e., one (and at most one) packet can cross an edge in each direction in one step.

phase 1: Each packet is routed to a randomly selected node in its column.

phase 2: Each packet is routed within its current row to its correct column.

phase 3: Each packet is routed within its correct column to its correct destination.

Assume that a phase starts after the previous phase ends. Contention resolution is handled using the farthest-first protocol in all phases except phase 2. In phase 2, assume that a packet which most recently entered the node has the highest priority (this simplifies the analysis, since this implies that once a packet starts moving in a row it is never delayed until it reaches its correct column.)

Your task is to show that the above protocol routes an arbitrary permutation in $O(\sqrt{N})$ steps using only queues of size $O(\log N)$ with high probability.

(Hint: In phase 1, how many packets can end up in the same row?

In phase 2: Consider a packet in node (i, j) which has to move rightward to go to its correct column. How many packets can delay this packet?

Bound the queue sizes at the end of phase 1 and phase 2 with high probability.)