18.435/6.848 Theory of Parallel and VLSI Computation	Fall 1997

Problem Set #5

Due: Wed. November 19 — *Prof. Leighton*

Reading (for lectures 14-16): Sections 3.4 – 3.5 of text, but skip ** sections.

Problems (for lectures 14–16):

1. Let F denote the FFT matrix whose i,j entry is ω^{ij} for $0 \leq i,j < N$ where ω is a principal N'th root of unity and N is a power of 2. Let F^{-1} denote the matrix whose i,j entry is ω^{-ij}/N. Show that $\omega = 2$ is a principal N'th root of unity in the ring of integers $mod 2^N - 1$. Show that for some N, F^{-1} is *not* the inverse of F over this ring.

2. Given any one-to-one mapping π of 2^r inputs to 2^r outputs in an r-dimensional Beneš network (one input per level 0 node and one output per level 2^r node), prove that there is a set of node-disjoint paths from the inputs connecting input i to output $\pi(i)$ for $1 \leq i \leq 2^r$.

3. Draw edge disjoint paths for the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 8 & 1 & 3 & 4 & 7 & 6 & 2 \end{pmatrix}$ in a three-dimensional Beneš network. (3.101)

4. A list of numbers is bitonic if it either is a sequence of decreasing numbers until some minimum point after which it is a sequence of increasing numbers or if it increases to some maximum and then decreases. Show that a bitonic list is sorted with just one pass through the butterfly comparator circuit (starting with MSB first). (Hint: Use the 0-1 sorting lemma.)

5. Draw node-disjoint routing paths for the packing problem in a three-dimensional butterfly where packets are initially contained in rows 101, 110, 011, and 100. Also, give an example showing that the greedy approach to packing fails if the levels of the butterfly are traversed in the opposite order (i.e., from level 1 to $\log N$ instead of level $\log N$ to 1). In other words, find an example where the greedy paths are not node-disjoint. (3.200,3.201)

6. Consider an arbitrary many-to-one N-packet routing problem on an N-input butterfly where each input node starts with one packet, and where packets destined for the same output can be combined. Show how to route all the packets in $O(\log N)$ steps in a normal fashion, given sufficient off-line precomputation. Extend this result to include one-to-many packets. You may assume that $m_1 + m_2 + \cdots + m_N = O(N)$, where m_i is the number of copies of the ith packet that need to be made $(1 \leq i \leq N)$. Lastly, extend the result to routing N-packets in $O(\log N)$ steps on an N-node shuffle-exchange graph. Can you do this with a normal algorithm? (3.213,3.214)

*7. Consider an N-node hypercube where every node has the capacity to handle a queue of up to k packets. Initially every node of the hypercube has between 0 and k packets, and there are N packets overall. Show how to redistribute the packets in $O(k + \log N)$ steps with local control so that every node has precisely one packet at the end. This is known as the *token distribution problem.* Note that packets do not have addresses or destinations in this problem, and each processor can process a packet on each of its edges at every step. (3.208)

*8. Given any comparison-based sorting algorithm for N items, show how to extend it to an algorithm for sorting kN items for any k by replacing each item with a sorted list of k items and each compare operation with a merge/split operation on lists of size k. More precisely, we replace any operation of the form $u = max(x, y), v = min(x, y)$ with $U =$ largest k elements in $U \cup Y$, V =smallest k elements in $X \cup Y$ where U is the k element list corresponding to u, and V is the k element list corresponding to v, and so forth.

$$ 9. A network with N inputs and N outputs is said to be *rearrangeable* if for any one-to-one mapping π of inputs to the outputs, we can construct edge-disjoint paths in the network linking the ith input to the $\pi(i)$th output for $1 \le i \le N$. Show that if every edge of the hypercube is doubled, and if one input and one output is attached to every node, then the network is rearrangeable. (3.108)

10. Consider a random routing problem on an m-sided k-dimensional array in which each processor starts with p packets. Show that if each packet is routed dimension-by-dimension to its destination, then the congestion across any edge is at most $O(pm + k \log m)$ with high probability. (For example, the packet starting at node $(x_1, x_2, \cdots, x_k)$ and ending in node $(y_1, y_2, \cdots, y_k)$ is routed through dimension 1 from $(x_1, x_2, \cdots, x_k)$ to $(y_1, x_2, \cdots, x_k)$, then through dimension 2 to $(y_1, y_2, \cdots, x_k)$ and through dimensions $3 \cdots k$ in order until it reaches $(y_1, y_2, \cdots, y_k)$.)

$$ 11. You have $\log N$ packets at each input of a butterfly, each with a random destination. The packets are routed using the greedy algorithm but cannot advance if the constant-size queue at the next node is full. How long does it take to route all the packets on average?

12. In *Radix Sort*, the items to be sorted are sorted in rounds starting with the least significant bit and progressing to the most significant bit. In each round, the relative order of items with the same bit is unchanged. Show how to radix sort N k-bit numbers in $O(k \log N)$ bit steps on a $\log N$-dimensional butterfly for $k \le O(\log N)$

*13. Given a random $2N$-packet circuit switching problem on a $\log N$ dimensional butterfly, it is shown in section 3.4.9 of the text that the greedy algorithm is likely to route $\Theta(N/\log N)$ packet paths to their destinations. Show that if we run the algorithm over again T times, each time eliminating packets (and their paths) that were previously successful and retrying packets that have not yet been successful, then every packet will eventually be successful for $T = O(\log N)$ with high probability.