Problem Set 2 (Due October 11, 2000)

Problem 1.

Give an algorithm for broadcasting an one-word message on the following networks. You may assume that links can transfer a whole word in a single cycle/clock-tick. What is the corresponding running time? Use $O(.)$ notation to express time bounds.
(i) n-cell linear array. Message is held by leftmost processor.
(ii) n-leaf complete binary tree. Message is held by root.
(iii) $(n \times n)$-cell 2d-array. Message is held by top-leftmost processor.

Problem 2.

You are given a hypercube with N nodes where $N = 2^{3n}$. Describe a matrix multiplication algorithm of two $M \times M$ matrices A and B on the hypercube, where $M = 2^n = N^{1/3}$. You need to explain which processor holds each element of A and B, where each computation is performed, and where each element of $C = A \times B$ is stored at the completion of the algorithm. (Hint: Use a triplet (a, b, c) to represent a processor id).

Problem 3.

Show how to perform parallel prefix on a butterfly (ie indicate how the PRAM algorithm presented in class can be made to work for a butterfly). If it helps, present the $N = 8$ case as an example.

Problem 4.

Construct a 5-bit Gray code.

Problem 5.

How many disjoint s-dimensional hypercubes are contained in an r-dimensional hypercube for $r \geq s$? (For example, a two-dimensional hypercube contains two one-dimensional hypercubes.)

Problem 6.

Show that r perfect outshuffles return a deck of n cards, where n is even, to its original order, if and only if r is the smallest integer such that $2^r = 1 \bmod (n-1)$. How many shuffles are needed when $n = 2^k$? How many for a regular deck ($n = 52$)? How many for a regular deck with two jokers ($n = 54$)? (Suggestion: Read section 3.3.2 if you know nothing of decks of cards. This problem is a combination of Problems 3.153, 3.154 and 3.155).

Problem 7.

Show how to perform a parallel prefix operation on the n nodes of a wrapped $\log n$-dimensional butterfly (page 445) in $\lg n$ steps (instead of the usual $2 \lg n$ steps) [Problem 3.199].

Problem 8*.

Given a $\sqrt{n} \times \sqrt{n}$ matrix of inputs $X = (x_{ij})$, a two-dimensional FFT on X consists of computing an FFT on the $\sqrt{n}$ values in each column of X, and then computing an FFT on the $\sqrt{N}$ values in each row of the resulting matrix.. Show how to compute a two-dimensional FFT in $\lg n$ steps with one pass through a $\lg n$-dimensional butterfly when n is a power of four. How fast can a two-dimensional FFT be computed on an n-node hypercube? [Problem 3.344]

Problem 9*.

Show that the N-node hypercube can be bisected by removing $O(N/\sqrt{\lg N})$ nodes, or by removing $N/2$ edges. (Stirling's approximation formula for the factorial is $n! = (n/e)\sqrt{2\pi n}$. Review the proof of Lemma 1.6, page 165, if you are not familiar with this formula). Hint: Solve Problem 3.3 first.

Problem 10*.

Show that a complete binary tree of $2n - 1$ nodes (i.e. n leaves) cannot be embedded with dilation one into a 2d-array with at least $2n - 1$ nodes, $n > 16$.

Hint: Show that the number of nodes in the array within distance k from some array node is at most $2k^2 + 2k + 1$ which is less than than $2^{k+1} - 1$, $k > 4$, the number of nodes within the same distance in the tree.