CS 1510 Midterm 4
Fall 2003

1. (20 points)

   (a) What is the most important reason that it is standard practice to ignore multiplicative constants when computing running times of algorithms/programs?

   HINT: Your answer should explain why we are not more precise, for example including multiplicative constants and ignoring low order terms, or less precise, for example ignoring poly-log factors.

   (b) Explain how to compute the minimum of n numbers $x_1, \ldots, x_n$ in $O(\log n)$ time with $n/\log n$ processors on a EREW PRAM. That is you want $T(n, p = n/\log n) = O(\log n)$.

   (c) What is the efficiency of the EREW algorithm in part (a). Start with the definition of efficiency.

   (d) Explain how to compute the minimum of n numbers $x_1, \ldots, x_n$ in $O(1)$ time with n^2 processors on a common CRCW PRAM.

2. (20 points) The input to this problem is a string C of n integers. The problem is to find the largest k, $0 \leq k < n$, such that

$$C[1]C[2]\ldots C[k] = C[n-k+1]\ldots C[n-1]C[n]$$

   That is, k is the length of the longest prefix that is also a suffix. So for example, if $C = 1, 2, 3, 4, 5, 6, 1, 2, 3$, then $k = 3$. Give a EREW parallel algorithm that runs in $O(\log n)$ time with a polynomial number of processors.

3. (20 points) Design a parallel algorithm that finds the maximum number in a sequence $x_1, \ldots, x_n$ of (not necessarily distinct) integers in the range 1 to n. Your algorithm should run in constant time on a CRCW PRAM with n processors.

   Note that it is important here that the x_i's have restricted range.