Lindsey Bieda and Joe Frambach
Parallel Problems
11.09.2011

9. The input to this problem is a character string C of n letters. The problem is to find the largest k 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. Give a EREW parallel algorithm that runs in poly-logarithmic time with a polynomial number of processors.

10. The input to this problem is a character string C of n letters. The problem is to find the largest k 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. Give a CRCW parallel algorithm that runs in constant time with a polynomial number of processors.

We will need $n/2$ processors in order to determine if there is a prefix and a suffix at most of length $n/2$. This is the maximum length to consider since any prefix longer than $n/2$ has less than $n/2$ as the remaining possible suffix and therefore the prefix and suffix cannot be equivalent. Each of these $n/2$ processors check for a prefix and suffix that is of length $\frac{n}{2}, \frac{n}{2}-1, \frac{n}{2}-2, \ldots, 1$. Since we write 0 initially to the solution location there is no need to verify. For each of these processors they check if the string at the beginning and the end of the given length is equal and then write to the memory location if and only if the k value stored there is less than the value being written.

11. Design a parallel algorithm for adding two n-bit integers. You algorithm should run in O($log\ n$) time on a CREW PRAM with n processors.

NOTE: If your algorithm is EREW, you might want to rethink since I don't know how to do this easily with out CR.

HINT: Use divide and conquer and generalize the induction hypothesis.