Homework 4

1. We have seen how to schedule the DAG corresponding to the standard algorithm for multiplying two $n \times n$ matrices in $O(\log n)$ time using $n^3$ processors. What is the optimal schedule for an arbitrary number $p$ of processors, where $1 \leq p \leq n^3$? What is the corresponding parallel complexity?

2. An item $X$ is stored in a specific location of the global memory of an EREW PRAM. Show how to broadcast $X$ to all the local memories of the $p$ processors of the EREW PRAM in $O(\log p)$ time. Determine how much time it takes to perform the same operation on the CREW PRAM.

3. Develop an optimal nonrecursive prefix-sums algorithm that is similar to the nonrecursive prefix-sums algorithm presented in class but that does not use the auxiliary variables $B$ and $C$. The input array $A$ should hold the prefix sums when the algorithm terminates.

4. Suppose we are given a set of $n$ elements stored in array $A$ together with an array $L$ such that $L(i) \in \{1, 2, \ldots, k\}$ represents the label of element $A(i)$, where $k$ is a constant. Develop an optimal $O(\log n)$ time EREW PRAM algorithm that stores all the elements of $A$ with label 1 into the upper part of $A$ while preserving their initial ordering, followed by the elements labeled 2 with the same initial ordering, and so on.

5. (Segmented Prefix Sums) We are given a sequence $A = (a_1, a_2, \ldots, a_n)$ of elements from a set $S$ with an associative operation $\ast$, and a Boolean array $B$ of length $n$ such that $b_1 = b_n = 1$. For each $i_1 < i_2$ such that $b_{i_1} = b_{i_2} = 1$ and $b_j = 0$ for all $i_1 < j < i_2$, we wish to compute the prefix sums of the subarray $(a_{i_1+1}, \ldots, a_{i_2})$ of $A$. Develop an $O(\log n)$ time algorithm to compute all the corresponding prefix sums. Your algorithm should use $O(n)$ operations and should run on the EREW PRAM.

6. Consider a cycle $C = (v_1, v_2, \ldots, v_n)$ with an additional set $E$ of edges between the vertices of $C$ such that, for each vertex $v_i$, there exists at most one edge in $E$ incident on $v_i$. Consider the problem of determining whether or not it is possible to draw all the edges in $E$ inside the cycle $C$ without any two of them crossing. Develop an $O(\log n)$ time algorithm to solve this problem. The total number of operations used must be $O(n)$. Your algorithm should run on the EREW PRAM model.