# CIS 786 Homework 1 (Due: Sep 23, 2004)

**Problem 1.** (20 points)

Go to the course web-page, handouts section, fill-in the linked form and submit the form.

**Problem 2.** (10 points)

How fast can you evaluate in parallel $x^n$ for some input $x$ and integer $n$? Express running time, number of processors used and parallel work in terms of $n$. Your algorithm must minimize all three parameters (for you to receive full credit).

**Problem 3.** (20 points)

In this problem we parallelize a "slow" sequential sorting algorithm to obtain superlinear speedup (of more than $p$).

**Step 1.** Our sequential algorithm is going to be one of the $O(n^2)$ sorting algorithms of the selection-sort, bubble-sort type. The input is $n$ keys $\langle x_1, \ldots, x_n \rangle$. Call this algorithm Alg1. Assume its running time is $n^2$.

**Step 2.** We show a modification of the sequential algorithm, called Alg2 (assume $n$ is multiple of $p$).

```
 Alg2 (x1..xn)
1. Split x1..xn into p equally sized groups, each one consisting of n/p keys.
2. Call Alg1 on each one of the p subsequences.
3. Merge the p sorted (after line 2 is executed) subsequences of step 2.
```

Explain how to perform in detail and efficiently step 2 and then analyze the running time of each of the three steps of Alg2. What is the total running time? Express result in asymptotic notation. For what values of $p$ is the running time $o(n^2)$? (this is a little oh not a Big oh!).

**Step 3.** We offer a glimpse on the parallelization of Alg1 (or Alg2). This is Alg3.

```
 Alg3 (x1..xn)
1. Processor 0 that has the input, splits x1..xn into p equally sized groups
   each one consisting of n/p keys and assigns the i-th group to processor i.
2. Processor i sorts the i-th group using Alg1.
3. The  p sorted subsequences  are sent to processor 0 which merges them.
```

Assume that a shared memory computer (eg. PRAM) is used and communication (eg. transmission and receipt of data are done through the shared memory) at unit cost. For an efficient implementation of all three steps, what is the parallel running time, processor usage and speedup for an efficient implementation of all three steps of Alg3. For what values of $p$ is the speedup strong superlinear (i.e. $\omega(p)$)? ($\omega$ is not $\Omega$). Examine separately the case $p = \Theta(\sqrt{n})$).

**Problem 4.** (20 points)

We have an array of $n$ elements. Attached to its element is a tag with a value of 0 or 1. We want to group the elements in the array in such a way that the 0-tagged elements come before the 1-tagged ones and the order of same-tagged elements must be preserved. So if element A is in position 5 and B is in position 8 and both A and B are tagged 0 we want in the output $A$ to precede $B$. Give an $O(\lg n)$-time algorithm that uses $n$ processors on an EREW PRAM. Can you solve the problem with only $n/\lg n$ processors?

**Problem 5.** (10 points)

What is an algorithm for Parallel_Sum if we have $p$ processors, where $p < n$ ? (You may assume that $n$ is a multiple of $p$).

**Problem 6.** (20 points)

Computer $F_n$, the nth Fibonacci number, given an integer $n$ as input. Show how to solve this problem in time $O(\lg n)$ on an EREW PRAM with $n$ processors. Assume that one word of memory is long enough to hold $F_n$. All arithmetic operations (add, subtract, multiply) take constant time. Recall that $F_n$ is defined by the following recurrence: $F_0 = 0, F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.