**Parallel Algorithms (15-492), Fall 07**
**Assignment #2** **Due: September 20th**

---

**Submission Instructions.** Since this is a programming assignment, you will hand in your code electronically. To hand in your solutions, please copy your files to

`/afs/cs.cmu.edu/academic/class/15492-f07/handin/`*username*`/assn2/`

where *username* is your Andrew ID.

---

## Problem 1: Scan

- Implement your own version of the NESL `pack` function using `plus_scan` with work complexity $O(n)$ and depth $O(\log n)$, or better. Assume that it only needs to work on vectors of integers. You can also assume that `plus_scan` has work $O(n)$ and depth $O(\log n)$.
- As we saw in class, the scan operation can be used to compute solutions to recurrence relations. In this question we will use scan to compute the Fibonacci sequence. Recall that the Fibonacci sequence is given by $F_1 = F_2 = 1$ and $F_{i+2} = F_i + F_{i+1}$. Write a NESL function `fib(n)` that computes the sequence [$F_1$, $F_2$, $\ldots$, $F_n$] using the scan operation. For example,

```
   fib(6)
⇒  it = [1, 1, 2, 3, 5, 8] : [int]
```

  The function should have work complexity $O(n)$ and depth $O(\log n)$. You may find the NESL function `scan` useful.

## Problem 2: Stock Market

Given the price of a stock at each day for $n$ days, we want to determine the biggest profit we can make by buying one day and selling on a later day. For example:

```
   stock([12, 11, 10, 8, 5, 8, 9, 6, 7, 7, 10, 7, 4, 2])
⇒  it = 5 : int
```

since we can buy at 5 on day 5 and sell at 10 on day 11. This has a simple linear time serial solution. Write a NESL program to solve the stock market problem with work complexity $O(n)$ and depth $O(\log n)$.

## Problem 3: Magic Pointers

A *pointer sequence* is an integer sequence in which each position is interpreted as a pointer to another position in the sequence. For example, the sequence `[1, 6, 2, 6, 2, 1, 6, 2]` represents two trees with roots at 2 and 6. The sequence `[1, 6, 5, 2, 0, 3, 4]` represents two cycles ($0 \to 1 \to 6 \to 4 \to 0$ and $2 \to 5 \to 3 \to 2$).

Please write an algorithm in NESL for the following problems. For parts a), b), and c), your algorithms should take $O(n \log n)$ work and $O(\log n)$ depth.

a) Detect if a pointer sequence has any cycles

b) For every node in a tree, return the root of the tree

c) Given a pointer sequence which only has cycles, return the number of cycles.
(*Hint:* use the element with maximum index.)

d) Improve your algorithm in part c) so that it takes expected $O(n)$ work and $O(\log n)$ depth.