ICS 443: Parallel Algorithms
Homework 5

Due: Wednesday, November 8, 2017, 9am

Instructions: You may discuss the problems with other students in the class, but you must write up the solutions on your own and give credit to the students with whom you discussed each problem.

1 Reversing a linked list (30 pts)

Describe a parallel algorithm that transforms a singly-linked list into a doubly-linked list. That is, given a linked list with only `next` pointers, describe how you would compute the `prev` pointers. Write down the pseudocode and analyze the time and work complexity. Your algorithm should run in $O(1)$ time and $O(n)$ work in the CREW PRAM model.

2 Pointer Hopping Algorithm (40 pts)

In lecture we have seen the following list ranking algorithm:

```
1:  function PointerHopping(v[1...n])          ▷ v is an array of vertices that defines the linked list
2:      for i = 1 to n in parallel do                                          ▷ Initialize all keys
3:          v[i].key ← 1
4:      end for
5:      for i = 1 to n in parallel do
6:          while v[i] ≠ NIL and v[i].next ≠ NIL do
7:              v[i].key ← v[i].key + v[i].next.key
8:              v[i].next ← v[i].next.next
9:          end while
10:     end for
11: end function
```

(a) (20 pts) Using loop invariant, prove that this algorithm computes the rank of every vertex correctly.

(b) (20 pts) Show that this algorithm runs in $O(\log n)$ time and $O(n \log n)$ work.

3 Size of randomized independent set (30 pts)

In lecture we have seen the following randomized algorithm for selecting an independent set in a linked list:

```
1: function IndependentSet(v[1...n])
2:     for i = 1 to n  in parallel do
3:         v[i].bit ← RandomBit()                    ▷ assign either 0 or 1 uniformly at random
4:     end for
5:     for i = 1 to n  in parallel do
6:         if i < n and v[i].bit = 1 and v[i].next.bit ≠ 1 then
7:             v[i].IndepSet ← true
8:         else
9:             v[i].IndepSet ← false
10:        end if
11:    end for
12: end function
```

Prove that the expected number of vertices whose *IndepSet* field set to true is $n/4$. *Hint: Use indicator random variables.*