Problem Set 3

This problem set is due in class on October 26th.

When a problem asks to give an algorithm, in your solution: (i) describe shortly and informally the main ideas in your solution; (ii) give a detailed description of the algorithm, using a style similar to (but possibly more concise than) the pseudo-code of CLR; (iii) prove the correctness of the algorithm; (iv) prove a bound on the time complexity of the algorithm.

Problem 1. Amortized analysis

Recall the problem of incrementing a counter (CLR p. 358). Using amortized analysis it can be shown that the worst-case time for a sequence of $n$ INCREMENT operations on an initially zero counter is $O(n)$ and thus the amortized cost of each operation is $O(1)$.

Now suppose we wish not only to increment a counter but also to reset it to zero (i.e. make all bits in it 0). Show how to implement a counter as a bit vector so that any sequence of $n$ INCREMENT and RESET operations takes time $O(n)$ on an initially zero counter.

(Hint: Keep a pointer to the high order 1).

Problem 2. Fibonacci Heaps

The height of $n$-node Binomial Heap is always $O(\log n)$. Show that this is not the case for Fibonacci Heaps by exhibiting, for any positive integer $n$ a sequence of Fibonacci-heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of $n$ nodes.

Problem 3. Graphs

Given a graph $G = (V, E)$ with $|V| = n$ (you can assume $V = \{1, \ldots, n\}$) 

(a) Present an algorithm that given $S_1, S_2 \subseteq V$, outputs $S_1 \cap S_2$ in time $O(n)$

(b) Present an algorithm that receiving as input the graph $G$ outputs a 4-cycle (if any exists) in time $O(n^3)$