Due Monday, 12/12/2016

Amortized Analysis Homework

All on paper, for the Instructor

(15 pts) Consider a queue data structure, where the two operations of interest are enqueue (at the back of the queue) and dequeue (from the front of the queue). A queue is thus a FIFO (first in-first out) structure. Suppose we implement a queue by using two stacks, A and B, which support operations of push and pop, as follows:

```
enqueue(x):   push x onto stack A

dequeue:      if B is not empty
                  return x = pop B
              if B is empty
                  repeat until A is empty
                      x = pop A
                      push x onto stack B
                  return x = pop B
```

a) This implementation seems to correctly emulate a queue. Illustrate (by drawing the two stacks) how it works for the following sequence of enqueue and dequeue operations:

```
enqueue 5
enqueue 7
dequeue
enqueue 3
enqueue 6
enqueue 9
dequeue
dequeue
dequeue
dequeue
```

b) The actual cost of a push or a pop operation is 1 unit of work. What is the worst-case single operation possible after a sequence of n enqueue and dequeue operations, and what is its cost?

c) Explain how a simplistic worst-case analysis would lead to the conclusion that after n operations, $O(n^2)$ units of work would have been done.

d) Using the accounting method, assign the lowest (integer) amortized cost possible to the enqueue operation and to the dequeue operation so that after any sequence of n1 enqueues and n2 dequeues, $n1 \geq n2$, the amortized cost is $\geq$ the actual cost. Explain why this works.

e) Show that the total amortized cost, which is an upper bound for the actual cost, over a sequence of n operations is O(n).