ECS 60 Written Homework #2: Lists, Stacks, Queues, Deques (28 points) Fall 2015

Due Wednesday, October 7th, 4:00 pm in 2131 Kemper

1. (2 points) Let *A* be an array of size $n \geq 2$ containing integers from 1 to $n - 1$, inclusive with exactly one repeated. Describe a fast algorithm for finding the integer in *A* that is repeated. (Goodrich, p. 150)

2. (2 points) How likely is the worst case for searching a skip list to occur? (Drozdek, p. 132)

3. (10 points, 2 points each) Consider the move-to-front, transpose, count, and ordering methods for lists. You will often need four sentences for each part. (Adapted from Drozdek, p. 132)
   a. In what case is a list maintained by these methods not changed?
   b. In what case do these methods require an exhaustive search of lists for each search, assuming that only elements in the list are searched for?
   c. Discuss the efficiency of each method when implemented as array.
   d. Discuss the efficiency of each method when implemented as a singly linked list.
   e. Discuss the efficiency of each method when implemented as a doubly linked list.

4. (2 points) Suppose you have two nonempty stacks *S* and *T* and a deque *D*. Describe how to use *D* so that *S* contains all of the elements of *T* below all of its original elements, with both sets of elements in *S* still in their original order. For example, with top on left, *S = 3, 4, 2; T = 5, 1, 9;* new *S = 3, 4, 2, 5, 1, 9*. (Adapted from Goodrich, p 225)

5. (2 points) Propose a data structure that supports the stack push and pop operations, and a third operation findMin, which returns the smallest element in the data structure, all in O(1) worst-case time. (Weiss, p. 119)

6. (2 points) Describe how to implement a queue using two stacks. Analyze the running time of the queue operations. (Cormen p. 203)

7. (2 points) Describe how to implement a stack using two queues. Analyze the running time of the stack operations. (Cormen p. 204)

8. (6 points, 2 points each) Assume a Stack class has only the operations: push, pop (which returns the top element), and isEmpty(). Assume a Queue class has only the operations: enqueue, dequeue (which returns the front element), and isEmpty(). Write C++ code that will reverse the order of elements on stack *S* of ints. (Adapted from Drozdek, p. 166)
   a. Using only two additional stacks, and no additional variables.
   b. Using only one additional queue, and no additional variables.
   c. Using only one additional stack, and some additional ints.

Sources of questions:

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Introduction to Algorithms*, New York, New York, McGraw-Hill, 1990.

Adam Drozdek, *Data Structures and Algorithms in C++, Second Edition*, Pacific Grove, CA, Brooks/Cole, 2001.

Michael T. Goodrich, Roberto Tamassia, and David Mount, *Data Structures & Algorithms*, *Second Edition*, Hoboken, NJ, John Wiley & Sons, 2011.

Mark Weiss, *Data Structures and Algorithm Analysis in C++, Fourth Edition*, New York, NY, Pearson Education, 2014.