CAS CS 112 – Spring 2011
Practice Midterm Questions

Problem 1

Write a function **int findMin(Queue q)**, which takes a queue of integers as input, and returns the minimum element in the queue, without changing the order of the elements. You can assume that the queue is non-empty and implements the usual operations (enqueue(), dequeue(), isEmpty()).

Problem 2

Now do the same for a Stack: write a function **void findMin(Stack s)**.

Problem 3

State whether each of the following statements is true of false and briefly explain why:

1. An **append** operation, which adds a value to the back of a list, can be implemented in constant time if we keep a reference to the end of the list.

2. Quicksort always runs in $O(n\log n)$ while mergesort runs in $O(n\log n)$ in the average case.

3. Insertion sort and selection sort are faster than mergesort and quicksort on some inputs.

Problem 4

Consider the following algorithm for sorting an array: Create a priority queue. Then for each of n elements in the array, insert it into the priority queue. Then perform n deleteMins, storing the elements back into the array left-to-right. Argue whether or not this algorithm correctly sorts the input. What is the algorithm's asymptotic running time? (Take a look at this one before Tuesday's class).

Problem 5

Consider an implementation of quicksort which picks the pivot as the first element of the array. Draw a diagram to illustrate how this implementaion performs on the following array of numbers: $\{1, 8, 4, 3, 5, 9, 2, 6, 10, 3, 5, 8, 7\}$. At each iteration, show how the array is partitioned and the location where the pivot is fixed. You can assume that the recursion bottoms out when the size of the array is 3 or less (at which point another sort is used), and that elements equal to the pivot go to the left.

Problem 6

Assume you have linked lists implemented using the following singly linked classes.

```
public class Node
{
    int data;
    Node next;
}

public class List
{
    Node head;
    int length;
}
```

To practice for the exam, write both an iterative and a recursive method to remove the last element of the list. Your code needs to account for the special cases of an empty list (just return) and a list with a single element. Your List method should use the following signatures:

Iterative version: `void RemoveLast();`
Recursive version: `void RemoveLast(Node l);`

Problem 7

Identify three bugs in the ArrayBasedList method below that inserts an item into an array-based list in sorted order (3 pts per bug). As in class, assume the "a" member variable is the array that stores the items and the "size" member variable stores the number of elements. The fact that array overflow is not tested does not count as a bug. Non-bugs that you identify as bugs will count against you. Line numbers are indicated for your convenience.

```
(1) void BuggyInsert(Item i) {

(2)     int j, k;

        // Find the location of the first item larger
        // than i and store it in variable j.  If all
        // items are smaller than i, j gets value "size"
(3)     for (j = 0; j < size || a[j] < i; j ++)
(4)         ;

        // Slide items from position j through position
        // size-1 one to the right to make room.
(5)     for (k = size-1; k > j; k--) {
(6)         a[k] = a[k-1];
(7)     }

        // now put i into its correct place
(8)     a[j] = i;
(9) }
```

Problem 8

Be ready for questions which involve the use of Generic syntax, like converting a data structure storing integers to one which can store arbitrary (generic) Records.