CSC C78F 1999
Assignment 1

due: Wednesday, October 6, 1999 at 1:10pm

1. Consider the abstract datatype GROWING MATRICES. The objects of this abstract datatype are square matrices and the following operations can be performed on them:

   INITIALIZE(M): initialize M to be a 0×0 matrix.

   ACCESS(M,i,j): return the (i,j)'th element of M, provided it has been assigned a value; otherwise report an error.

   ASSIGN(M,i,j,x): assign the value x to the (i,j)'th element of M, provided M has an (i,j)'th element; otherwise report an error.

   GROW(M): add one more row and one more column to M.

   Describe two different data structures for the abstract datatype GROWING MATRICES, including algorithms for the operations. Be brief, clear, and precise in your explanations.

   What are the worst case running times of your algorithms? Justify your answers.

2. Consider the following algorithm to merge two linked lists that are in nondecreasing order into one list containing all of the elements arranged in nondecreasing order.
   Each list element consists of a value and a pointer to the next element in the list.
   The variables a and b point to the beginning of the input lists, c is a pointer to the beginning of the output list and d is a pointer to the end of the output list.

```
if a = nil then c ← b
else if b = nil then c ← a
else %Create a list with one dummy element.
    c ← (0, nil)
    d ← c
    loop
        if a.value < b.value
        then d.next ← a
        d ← a
        a ← a.next
        d.next ← nil
        if a = nil
        then %Concatenate the rest of b's list to the end of the output list.
            d.next ← b
            exit loop
    else %Move the first element of b's list to the end of the output list.
        d.next ← b
```

$d \leftarrow b$
$b \leftarrow b.next$
$d.next \leftarrow nil$
if $b = nil$
then %Concatenate the rest of a's list to the end of the output list.
    $d.next \leftarrow a$
    exit loop
end loop
%Remove the dummy element from the beginning of the list.
$c \leftarrow c.next$

Suppose that the lists a and b are disjoint and together they contain the values $1, 2, \ldots, n$. Suppose that each element is equally likely to be in either list.

(a) Assume that which list each element is in is independent of which lists the other elements are in. What is the expected number of operations (comparisons and assignments) performed by this algorithm? Justify your answer.

(b) Suppose that list a has $\lceil n/2 \rceil$ elements, which are equally likely to be any of the elements of $\{1, 2, \ldots, n\}$. What is the expected number of operations (comparisons and assignments) performed by this algorithm? Justify your answer.

3. (a) Draw the red-black trees that result from separately performing the operations INSERT 40 and DELETE 17 on the red-black tree in Figure 14.1 at the top of page 264 in CLR. Show your intermediate steps.

   (b) For every nonnegative integer k, what is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number? Justify your answers.

   (c) For every positive integer n, describe a red-black tree on n keys that realizes the largest possible ratio of red internal nodes to black internal nodes. What is this ratio? What tree has the smallest possible ratio, and what is the ratio? Justify your answers.