CSC C78F 2000
Assignment 1

due: Wednesday, September 27, 2000

1. Consider the abstract data type CACHE. An object of this abstract data type is a subset C of $\{1, ..., m\}$ of size at most k. Initially, $C = \emptyset$. The only operation is *access*, which takes a parameter i. It adds i to C and, if this causes the size of C to become bigger than k, removes the element from C that was least recently the parameter of an access operation.

   For example, if $k = 2$ and the sequence:

   $$access(1), access(3), access(2)$$

   or the sequence:

   $$access(3), access(1), access(3), access(2), access(2)$$

   is performed, then $C = \{2, 3\}$.

   Give TWO significantly different data structures for this abstract data type. Informally justify why your data structures (including the *access* algorithms) are correct. For each, give the worst case space complexity and worst case time complexity of the *access* operation as functions of k and m. Justify your answers.

2. Consider the following algorithm which takes as input a positive integer n and an array x containing n integers, all distinct.

   $y \leftarrow x[n]$
   $j \leftarrow 1$
   while $x[j] < y$ do $j \rightarrow j + 1$ end while
   write j

   (a) Give a decision tree for this algorithm for $n = 3$.

   (b) What is the worst case number of comparisons performed by this algorithm for an array of length n? Justify your answer.

   (c) Suppose n is fixed and all permutations of x are equally likely. What is the probability that the number of comparisons in (b) will occur? Justify your answer.

   (d) Suppose that $n = 3$ and the array contains the integers 1, 2, and 3. If all permutations are equally likely, what is the expected number of comparisons that will be performed? Justify your answer, including an explicit description of the probability space.

   (e) Suppose that $n = 3$ and the array contains the integers 1, 2, and 3. If $Pr[x[i] = i] = Pr[x[i] \neq i]$ for $i = 1, 2, 3$ and $Pr[x = 1, 2, 3] = .3$ what is the expected number of comparisons that will be performed? Justify your answer, including an explicit description of the probability space.

3. Consider the following simplified version of the Mastermind game.
   One player chooses a sequence of two (not necessarily different) colours from among {Blue, Green, Red}. The second player must discover this sequence.
   Each turn, the second player chooses a sequence of two (not necessarily different) colours from among {Blue, Green, Red}. Then the first player responds with the number of positions in which the two sequences agree.
   The game ends when the first player answers 2, indicating that the second player has discovered the sequence.

   (a) Give an efficient decision tree for playing this game. Justify your answer.

   (b) What is the number of turns taken by your algorithm, in the worst case? Justify your answer.

4. Consider a binary tree which, at each node v, stores a separate piece of information (for example, a flag) about each path from v going down to a leaf.

   (a) What is the minimum possible number of pieces of information stored in this tree, if it has height k?

   (b) What is the maximum possible number of pieces of information stored in this tree, if it has height k?

   Justify your answers.

5. In section 8.1 of the textbook, PARTITION partitions the array into two parts instead of three as described in class.

   (a) Explain how to modify RANDOMIZED-SELECT (defined in section 10.2) to use the class PARTITION algorithm.

   (b) Experimentally compare the running time of RANDOMIZED-SELECT (using PARTITION from the text book), RANDOMIZED-SELECT2 (using the class PARTITION algorithm with 2-way comparisons), and RANDOMIZED-SELECT3 (using the class PARTITION algorithm with 3-way comparisons).

   Specifically, you must do the following:

   - Implement RANDOMIZED-SELECT, RANDOMIZED-SELECT2, and RANDOMIZED-SELECT3 in either JAVA, C, or C++ and hand in properly documented printouts of your code.
   - Choose a good set of data on which to run these algorithms. Hand in a written description of your data and an explanation of why it is a good choice.
   - Hand in the results of your runs.
   - Draw an appropriate graph comparing the three programs.