Analysis of Algorithms: Assignment 4

Due date: March 1 (Wednesday)

Problem 1 (5 points)
Write an algorithm Shuffle(A, n) that randomly re-orders an array $A[1..n]$, that is, generates one of its permutations; the probability of all permutations must be the same. You may use the Random(p, r) procedure, which returns a random integer between p and r.

Problem 2 (3 points)
Using Figure 9.2 (page 176) in the textbook as a model, illustrate the operation of Counting-Sort on the array $A = \langle 6, 3, 4, 5, 6, 4, 3, 8, 4, 1, 2 \rangle$.

Problem 3 (2 points)
Give a version of the Enqueue and Dequeue procedures that check for the overflow and underflow errors. An *overflow* error is adding an element to the full queue, and an *underflow* is trying to get an element from an empty queue.

Problem 4 (bonus)
This is an optional problem, inherited from Exam 1, which allows you to get 2 bonus points toward your final grade. You cannot submit this bonus problem after the deadline.

Consider the problem of finding the kth smallest element of an array $A[1..n]$, that is, the element that would occupy the kth position after sorting the array. For example, if the array is $<6, 4, 8, 2, 10, 0>$ and $k = 3$, then the kth smallest element is 4, since it is the third element in the sorted array $<0, 2, 4, 6, 8, 10>$.

Write a modified version of Quick-Sort for finding the kth smallest element of a given array. Your algorithm should be *significantly more efficient* than the original Quick-Sort. Sorting the whole array and then returning the kth element is not an appropriate solution.