Analysis of Algorithms: Solutions 4

```
                      X
                      X
                  X X X
number of         X X X
homeworks     X X X X X
        X     X X X X X
        X     X X X X X
    X   X X X X X X X X
    ---------------------
    1 2 3 4 5 6 7 8 9 10
             grades
```

Problem 1
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.

SHUFFLE(A, n)
for $i \leftarrow n$ downto 2
    do $j \leftarrow$ RANDOM$(1, i)$
        exchange $A[i] \leftrightarrow A[j]$

Problem 2
Illustrate the operation of COUNTING-SORT on the array $A = \langle 6, 3, 4, 5, 6, 4, 3, 8, 4, 1, 2 \rangle$.

The values in array C are as follows:

	1	2	3	4	5	6	7	8
after first loop:	0	0	0	0	0	0	0	0
after second loop:	1	1	2	3	1	2	0	1
after third loop:	1	2	4	7	8	10	10	11
after fourth loop:	0	1	2	4	7	8	10	10

Problem 3

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.

ENQUEUE(Q, n, x)
if $head[Q] = tail[Q] + 1$ or $(head[Q] = 1$ and $tail[Q] = n)$
  then error "overflow"
$Q[\text{tail}[Q]] \leftarrow x$
if $tail[Q] = n$
  then $tail[Q] \leftarrow 1$
  else $tail[Q] \leftarrow tail[Q] + 1$

DEQUEUE(Q, n)
if $head[Q] = tail[Q]$
  then error "underflow"
$x \leftarrow Q[head[Q]]$
if $head[Q] = n$
  then $head[Q] \leftarrow 1$
  else $head[Q] \leftarrow head[Q] + 1$
return x

Problem 4

Write a modified version of QUICK-SORT for finding the kth smallest element of an array.

QUICK-FIND(A, p, r, k)
if $p = r$
  then return $A[p]$
$q \leftarrow$ PARTITION(A, p, r)
if $k \leq q$
  then return QUICK-FIND(A, p, q, k)
  else return QUICK-FIND$(A, q + 1, r, k)$