CS 610, Spring 2009, Prof. Calvin
Homework #4 Solutions

C-4.14 Express the number in base n.

C-4.19 Modify **mergesort** so that it returns not only a sorted array, but the number of inversions in the array. That way recursive calls to **mergesort** on each half of an array return the number of inversions in that half. Then we add these two quantities to the number of inversions that span the two halves. To accomplish this last part, we modify **merge**. When we choose an element from the right half in the merge, then there is an inversion for each of the elements remaining in the left half. We add these to the other two inversion counts to get the total.

C-4.23 Use QuickSelect to find the $(n + \lceil\sqrt{n}\rceil)/2$-st order statistic, and then the $(n - \lceil\sqrt{n}\rceil)/2$-st order statistic. The average time is linear.

C-4.27 Initially call $RandomizedQuantiles(A, k, 1, n)$, defined by:

RandomizedQuantiles(A, j, q, r)
  if $j > 1$
    $p \leftarrow RandomizedSelect(A, \lfloor j/2 \rfloor (r-q), q, r)$
    $RandomizedQuantiles(A, \lfloor j/2 \rfloor, q, p-1)$
    $RandomizedQuantiles(A, j - \lfloor j/2 \rfloor, p+1, r)$

Let $T(k, n)$ denote the expected running time. Then (ignoring integer rounding):

$$\begin{aligned}
T(k,n) &= \Theta(n) + 2T(k/2, n/2) \\
&= \Theta(n) + 2\left(\Theta(n/2) + 2T(k/4, n/4)\right) \\
&= 2\Theta(n) + 4T(k/4, n/4) \\
&\;\;. \\
&\;\;. \\
&\;\;. \\
&= j\Theta(n) + 2^j T(k/2^j, n/2^j).
\end{aligned}$$

With $j = \lg(k)$, we get

$$T(k,n) = \lg(k)\Theta(n) + kT(1, n/k) = \Theta(\lg(k)n).$$