# CPS 130 Homework 6
# Quicksort

due Mon May 27th

*Write and justify your answers in the space provided.*[1]

1. (CLRS 7.4-5) The running time of quicksort can be improved in practice by taking advantage of the fast running time of insertion sort when its input is "nearly" sorted. When quicksort is called on a subarray with fewer than $k$ elements, let it simply return without sorting the subarray. After the top-level call to quicksort returns, run insertion sort on the entire array to finish the sorting process. Argue that this sorting algorithm runs in $O(nk+n\lg(n/k))$ expected time. How should $k$ be picked, both in theory and in practice?

[1]Collaboration is allowed, even encouraged, provided that the names of the collaborators are listed along with the solutions. Students must write up the solutions on their own.

2. (CLRS 7-3) Professors Dewey, Cheatham, and Howe have proposed the following "elegant" sorting algorithm:

   Stooge-Sort$(A, i, j)$
   if $A[i] > A[j]$
       then exchange $A[i] \leftrightarrow A[j]$
   if $i + 1 \geq j$
       then return
   $k \leftarrow \lfloor (j - i + 1)/3 \rfloor$
   Stooge-Sort$(A, i, j - k)$
   Stooge-Sort$(A, i + k, j)$
   Stooge-Sort$(A, i, j - k)$

   **a.** Argue that Stooge-Sort$(A, 1, length[A])$ correctly sorts the input array $A[1..n]$, where $n = length[A]$.

   **b.** Give a recurrence for the worst-case running time of Stooge-Sort and a tight asymptotic ($\Theta$-notation) bound on the worst-case running time.

   **c.** Compare the worst-case running time of Stooge-Sort with that of insertion sort, merge sort, heapsort, sock sort, and quicksort. Do the professors deserve tenure?