1. Consider the following algorithm for finding the three smallest numbers, and their order, in a list (of size n): Take the first three numbers and sort them. Then, one at a time, take each remaining number compare it to the largest of the three, then the middle of the three if necessary, and finally the smallest of the three if necessary. Always retain the three smallest numbers (seen so far) in order.
   (a) Write the pseudo code for this algorithm.
   (b) Give the exact number of comparisons in the best case. Justify.
   (c) Give the exact number of comparisons in the worst case. Justify.
   (d) Give the exact number of comparisons in the average case. Justify. Write your answer using the harmonic sum H_n so that you will get a true exact answer.

2. (a) Assume you have an alphabet of letters from "p" though "t" plus "a". Illustrate the operation of radix sort on the following list of English words: star, tars, arts, rats, tsar, trap, tats, tarp, part, rasp.
   (b) Use "tat" and "rasp" in an English sentence that shows that you understand the meaning of both words.

3. We showed in class that a list of n numbers with range S can be sorted using radix sort in time $\Theta(\frac{n \log S}{\log n})$. This implies, for example, that if $S = n^4$, n numbers can be sorted in (linear) time $\Theta(n)$.

   Without relying directly on the above result, show that you can sort n numbers in the range $0, \ldots, n^4 - 1$ in (linear) time $\Theta(n)$. Your answer can be very brief. Just give enough detail so that it is convincing that you know how to do it.