- 1. Draw the Decision Tree for Bubble Sort on three elements A, B, C (which start in positions indexed by 1, 2, 3 of an array, respectively). Note that Bubble Sort is inefficient, so it does some redundant comparisons and some nodes will have only one child.
- 2. Assume that your computer has special hardware that finds the minimum of k (or fewer) elements in one comparison step. Your answers to this question should have n and k as parameters.
 - (a) Design an efficient algorithm based on Merge Sort to sort n elements using this special hardware. (This is an upper bound.)
 - (b) Analyze your algorithm. Get the high order term exactly.
 - (c) Use decision trees to find a lower bound for sorting when using this special hardware.
 - (d) Compare your upper and lower bounds.
- 3. (a) Assume you have an alphabet of letters from "o" to "u". Illustrate the operation of radix sort on the following list of English words:

tote, soup, soot, pout, toot, sups, tour, opts, rout, tors

(b) Use the words "sup" and "tor" in one English sentence that shows that you understand the meaning of both words.