Homework 5, CS420, Fall 2011

Last revised 10/11/11 11:50

Due 10/14 in class

1. 7.1-2, page 174, `Partition` with repeats.

2. 9.3-5, page 223, Selection with a black box

3. Finding the median of two sorted arrays of length n.

   Here is one way to define a median of a list of real values: it is an element x such that at least half of the elements (counting x) are less than or equal to x, and at least half (counting x) are greater than or equal to x. When all of the elements are unique, a list has one median if it has odd length and two if it has even length.

   a. Briefly describe how to get an $o(n \log n)$ time bound based on getting the elements into a single sorted list.

   Let us reduce the problem to the special case where n is $2^k + 1$ for some integer k, except when we're in a base case of $n = 1$. Let us do this by padding the beginning of one of the arrays with elements that are smaller than any of the input elements, and padding the end of the other with elements that are larger than any of the input elements.

   b. In terms of n, how many elements do we need to add to each array? Consider using logs and ceilings in your expression.

   Physically adding the new elements would take $\Theta(n)$ time, which would ruin the $o(n)$ time bound we will now go for. Convince yourself that this is unnecessary; instead, you can design the algorithm to "pretend" that they are there when you solve the problem by implicitly renumbering the positions in the arrays.

   For simplicity, assume that all of the elements are unique. Since $2n$ is even, there will be two medians of X and Y. Design your algorithm to return both of them. Since n is odd, each of X and Y has a unique median.

   c. Describe the inputs to your procedure, and tell how to find which element of X to compare with which element of Y. For each of the two possible outcomes, tell which elements of X and which elements of Y will make up your recursive call. Justify that the recursive call satisfies the precondition about each of the two sets having $2^k + 1$ elements.

   d. Describe your base case and how you will handle it.

   e. Illustrate the execution of your algorithm by telling the set of pairs of elements it compares when given the following two lists: $X = (1, 3, 4, 7, 8, 11, 13, 19, 24)$, $Y = (2, 5, 6, 12, 15, 16, 17, 21, 25)$.

   f. Give a recurrence and derive a big-Θ bound as a function of n.

4. 9.3-9, page 224, Oil of Olay

5. 9-4, page 226: Expected bounds for selection.
6. (Extra credit) 9-2, page 225, Weighted median, parts a,b, c.
7. (Extra credit) 9-3, Small order statistics.