CS 122a Fall 2010, HW 2, Due Monday October 10

This is a little longer, so you have a ten days to do it. But don't wait until the last minute to start it.

1. By the unwrapping method discussed in class, solve the following recurrence relation. Show your work.

$$T(n) = 5T(n/4) + 3n$$

Then solve it by using the Master Method, showing in detail which rule applies.

2. Apply the Master method (described in two postings on the class website), or, when possible, apply the material in sections 5.1 and 5.2 of the book, to solve each of the following recurrences, or state that the Master method does not apply. Justify your answers. Note that the Master method covers all the cases covered by section 5.2, and also covers some additional cases.

2a. $T(n) = 2T(n/2) + n^2$
2b. $T(n) = 2T(n/2) + 5$
2c. $T(n) = 4T(n/2) + n$
2d. $T(n) = 4T(n/2) + n^2$
2e. $T(n) = 4T(n/2) + n^3$

2. Do problem 3 on page 246. This may be a hardish problem.

3. Do problem 6 on page 248. This may be an easyish problem.

4. Suppose you have computed the number of inversions in a permutation Π of n integers. Let Π^r denote the reverse of permutation Π. For example $\Pi = 2, 3, 1$ then $\Pi^r = 1, 3, 2$. What is the most efficient way of computing the number of inversions in Π^r.

5a. Show how to Merge (not mergesort) three *ordered* lists each of size $n/3$, with a total of at most $\frac{5}{3}n$ compare operations.

The Mergesort algorithm is now shown next, where the parameter A in (A,n) is the list to be sorted, and the second parameter, n, is the length of the list.

Procedure mergesort(A,n);
If $n = 1$ then return A.
Else divide A into three equal sized lists B, C, D.
$B := mergesort(B, n/3);$
$C := mergesort(C, n/3);$

$D := mergesort(D, n/3);$
return merge(B,C,D);
end.

5b. Set up and solve a recurrence relation to analyze the worst-case number of compares that this version of mergesort does.