1. Give asymptotic upper and lower bounds for $T(n)$ for each of the following recurrences (use the Master Method when possible). Assume that $T(n)$ is constant for $n$ less than or equal to 2. Make your bounds as tight as possible, and justify your answers (master method should do that for you, when it is applicable). Show your work, and state the MM case that applies as well as computed values for constants.

(a) $T(n) = 9T(\frac{n}{4}) + n^2$
(b) $T(n) = 2T(\frac{n}{2}) + n \log^2 n$
(c) $T(n) = 4T(\frac{n}{2}) + n$
(d) $T(n) = 2T(n-1) + n^2$
(e) $T(n) = T(\frac{n}{2}) + n$
(f) $T(n) = T(\frac{n}{10}) + n$
(g) $T(n) = 2T(\frac{n}{2}) + \sqrt{n}$

2. This is a problem from the text.

Professors Howard, Fine, and Howard have proposed the following “elegant” sorting algorithm:

```
Stooge-Sort (A, i, j)
    if A[i] > A[j]
        then exchange A[i] and A[j]
    if i+1 >= j
        then return
    k = floor ( (j-i+1) / 3 )
    Stooge-Sort (A, i, j-k) // first two-thirds
    Stooge-Sort (A, i+k, j) // last two-thirds
    Stooge-Sort (A, i, j-k) // first two-thirds, again
```

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

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

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

3. Assume you have a list of $n$ real values. Devise an $O(n)$ algorithm to find a number that is not in the set. Argue that $\Omega(n)$ is a lower bound on the steps required to solve this problem.

4. Suppose that you have an $n$ by $n$ matrix in which the rows and columns are both sorted in ascending order, and you want to determine whether a particular value $X$ is in the matrix. Design and $O(n)$ algorithm to solve this problem.