1. Assume that you execute randomized selection trying to find the smallest value in a list (of size n).
   (a) Write a recurrence for the exact average number of comparisons.
   (b) Use constructive induction to get an upper bound the average number of comparisons. Get the exact high order term.

2. One way to find the median of a list is to sort the list and then take the middle element.
   (a) Assume you use Bubble Sort to sort a list with 9 elements (i.e. $n = 9$). Exactly how many comparisons do you use (in the worst case)?
   (b) i. Assume you use Mergesort to sort a list with 9 elements (i.e. $n = 9$). Exactly how many comparisons do you use (in the worst case)?
       ii. We know that if n were a power of 2 then the number of comparisons would be $n \lg n - n + 1$. What is this value for $n = 9$ rounded to the nearest integer?
       iii. How do the two values above compare?

3. You can actually find the median by running a sorting algorithm and stopping early, as soon as you know the median.
   (a) Assume you use Bubble Sort to find the median of 9 elements (i.e. $n = 9$), but stop as soon as you know the median. Exactly how many comparisons do you use (in the worst case)?
   (b) Assume you use Mergesort to find the median of 9 elements (i.e. $n = 9$), but stop as soon as you know the median. Exactly how many comparisons do you use (in the worst case)?

4. The selection algorithm (to find the kth smallest value in a list), described in the class (and in the book), uses columns of size 5. Assume you implement the same selection algorithm using columns of size 9, rather than 5.
   (a) Exactly how far from either end of the array is the median of medians guaranteed to be. Just give the high order term. (Recall that with columns of size 5 we got $\frac{3n}{10}$.)
   (b) It turns out that there is an algorithm that finds the median of 9 elements with 14 comparisons. Using this algorithm, briefly list each step of Selection with columns of size 9 and how many comparisons the step takes. Note that partition can now be done with only $(4/9)n$ comparisons; use this value in your analysis.
   (c) Write a recurrence for the number of comparisons the algorithm uses.
   (d) Solve the recurrence using constructive induction. Just get the high order term exactly.
   (e) With more careful analysis, in class we could have obtained $16n$ comparisons using columns of size 5. How does this new value, using columns of size 9, compare?