1. (60 points) Professor Sazawal tells you excitedly that she has an improvement on Counting Sort that sorts n integers in the range $0, \ldots, k-1$ in time $\Theta(n + \sqrt{k})$. She then asks you what improvement you can get on Radix Sort using her wonderful algorithm.
   (a) Let n be the number of elements to be sorted, r the radix, and d the number of digits. Using Professor Sazawal's version of Counting Sort, what is the time for Radix Sort in terms of n, r, and d?
   (b) Assume all of the elements are integers in the range $0, \ldots, S-1$, what is the relationship between r, d, and S?
   (c) Rewrite the running time of the new Radix Sort in terms of n, r, and S.
   (d) Find the optimal value for r (in terms of n and S). Show your work. You should get the high order term for r exactly.
   (e) Rewrite the running time of the new Radix Sort in terms of n and S, using your optimal value of r.
   (f) What should you tell Professor Sazawal? Your answer should be brief, useful, cogent, and cover the important points!
   (g) BTW. Do you believe that Professor Sazawal has actually found such an algorithm? Why or why not?

2. (40 points) Assume you use the selection algorithm from class (and from CLRS), but columns of size 11 (instead of 5). Assume we use *bubble sort* to find the median of each column.
   (a) Briefly list each step that the algorithm actually computes and how many comparisons the step takes.
   (b) Write a recurrence for the number of comparisons the algorithm uses.
   (c) Solve the recurrence using constructive induction. Just get the high order term exactly.
   (d) How does this value compare to what we got in class with columns of size 5?