ECS222a Graduate Algorithms
Homework 2

You are encouraged to talk to other people about these problems, but please **write up the solutions by yourself**. Always explain the answer in **your own words**; do not copy text from the book, other books, Web sites, your friends' homework, your friend's homework from last year, etc. If you use other books, Web sites, journal papers, etc. to get a solution, cite the reference and explain the solution in your own words, so that we can tell that you understand the material you are using. Always explain your solution as you would to someone who does not understand it, for instance to a beginning graduate student or an advanced undergraduate.

Please type your homework. If you know LaTeX, use that. If not, you may ype your answers in any word processing system and write in mathematical notation by hand as necessary. Include pictures if appropriate; you can draw in pictures by hand or include them in the file.

1. Prove that any comparison-based algorithm, randomized or deterministic, for finding the median of n numbers must make $\Omega(n)$ comparisons.

2. Do problem 12-2 (A radix tree is the same as a trie). Does the lower bound from Section 8.1 apply to this sorting algorithm? Explain why or why not.

3. In this problem we will show that LZ compression cannot compress a random string very much; getting good compression depends on finding repetition in the string, and random strings don't repeat very much. We consider compressing a long random string S of length n on the alphabet $\{0, 1\}$; each character is chosen independently at random with equal 0 and 1 having equal probability.
   a) First we argue that it is very unlikely that any long substring is repeated within S. Find a function $k(n)$ for which you can show that

   $$\Pr[\text{ any substring of length } k(n) \text{ is repeated within } S] = o(1/n^2)$$

   We do not count two overlapping copies of a substring as repeated; for example in the string "01010" we count "01" as repeated but not "010". To get the best lower bound you can on the compression factor, you will want to get the asymptotically smallest k you can.
   b) Argue that if no substring of length $\geq k(n)$ is repeated within S then the maximum length of a path in the LZ trie built when compressing S is at most $k(n)$.
   c) Argue that if the maximum length of a path in the trie is at most $k(n)$, then the LZ algorithm presented in class produces a string of length $\Omega(n/k(n))$.

4. Do problem 11-4, parts a and b.

5. Consider hashing n items into a hash table of size exactly n. Assume (unrealistically) that our hash function $h(a)$ is perfectly random; it places item a into a random bucket, independent of where all other items are placed. We will argue that the most items in any bucket is $O(\frac{\ln n}{\ln \ln n})$ with probability at least $1 - O(1/n)$.

a) Fix a single bucket. Let q be the probability that k items hash to this bucket. Express q as a function of k and n.
b) Get an upper bound on q using inequality (C.5) on page 1097; this is a very handy inequality and you probably want to make a note of it.
c) Show that for $k \geq \frac{2e \ln n}{\ln \ln n}$, we have $q = o(\frac{1}{n^2})$. You will probably want to use part b), and it will probably be helpful to notice that

$$(\ln n)^{\frac{\ln n}{\ln \ln n}} = ((\ln n)^{\frac{1}{\ln \ln n}})^{\ln n} = (e^{\frac{\ln \ln n}{\ln \ln n}})^{\ln n} = n$$

d) Argue that the probability that any bucket contains more than $\frac{2e \ln n}{\ln(\ln n)}$ items is $O(1/n)$.