CS 561, HW 3

Prof. Jared Saia, University of New Mexico

1. You are on an island where each inhabitant is either a knight or a knave. When asked a question, knights always tell the truth, but knaves may either lie or tell the truth. On this island, you can only ask questions of the form: "Hey Person X: what is Person Y's type?". Each person always answers a question about someone else in the same way, so there's no reason to ask the same question twice.

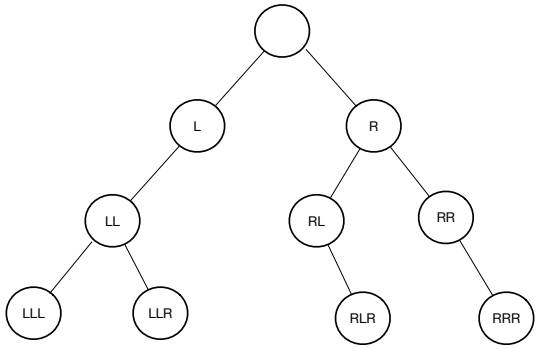
   (a) Assume there are n people on the island, and a strict majority of them are knights. Describe an efficient algorithm to identify the type of each person. Hint 1: You can do this in $o(n^2)$ questions. Hint 2: First try to identify 1 knight. Use recursion!

   (b) Prove that if at most half the inhabitants are knights, then it is impossible to solve the problem.

2. A product of matrices is fully parenthesized if it is either a single matrix, or the product of two fully parenthesized matrix products, surrounded by parenthesis. For example $(A_1((A_2A_3)A_4))$ is fully parenthesized. Prove by induction that any full parenthesization of a product of n matrices has exactly $n-1$ pairs of parenthesis.

3. Consider a rooted binary tree with nodes are labelled as follows. The root node is labelled with the empty string. Then, any node that is a left child of a node with name σ receives the name σL and any node that is the right child of that node receives the name σR.

   Give a recurrence relation returning the number of R's in all labels of all nodes. For example, the following tree has 10 R's.



Hint: For a node v, let $f(v)$ be the number of R's in the tree rooted at v, if the naming started at v. Also, let $\ell(v)$ (resp. $r(v)$) be the left (resp. right) child of v if it exists or NULL otherwise. Finally, let $s(v)$ be the number of nodes in the subtree rooted at v and assume this value is stored at each node. Now give a recurrence relation for $f(v)$.

4. **Primes and Probability**. In this problem, you will use the following facts: (1) any integer can be uniquely factored into primes; (2) the number of primes less than any number m is $\Theta(m/\log m)$ (this is the prime number theorem).

   We will also make use of the following notation for integers x and y: 1) $x|y$ means that x "divides" y, which means that there is no remainder when you divide y by x. and 2) $x \equiv y \pmod{p}$ means that x and y have the same remainder when divided by p, or in other words, $p|(x-y)$.

   (a) Show that for any positive integer x, x factors into at most $\log x$ unique primes. Hint: 2 is the smallest prime.

   (b) Let x be a positive integer and let p be a prime chosen uniformly at random from all primes less than or equal to m. Use the prime number theorem to show that the probability that $p|x$ is $O((\log x)(\log m)/m)$.

   (c) Now let x and y both be positive integers less than n, such that $x \neq y$, and let p be a prime chosen uniformly at random from all primes less than or equal to m. Using the previous result, show that the probability that $x \equiv y \pmod{p}$ is $O((\log n)(\log m)/m))$.

   (d) If $m = \log^2 n$ in the previous problem, then what is the probability that $x \equiv y \pmod{p}$. Hint: If you're on the right track,

you should be able to show that this probability is "small", i.e. it goes to 0 as n gets large.

(e) Finally, show how to apply this result to the following problem. Alice and Bob both have large numbers x and y where x and y are both at most n, for n a very large number. They want to check to see if their numbers are the same, but Alice does not want to have to send her entire number to Bob.[1]
What is an efficient randomized algorithm for Alice and Bob that has "small" probability of failure? How many bits does Alice need to send to Bob as a function of n, and what is the probability of failure, where failure means that this algorithm says x and y are equal, but in fact they are different?

[1]For example, x and y represent large binary files (think terabytes), and Alice and Bob want to check that their files are equal, without Alice having to send her entire file to Bob