

1. Suppose that social security numbers were issued randomly, with replacement. That is, your social security number consists of just 9 randomly generated digits, and no check was made to ensure that the same number was not issued twice. Sometimes the last four digits of your social security number are used as a password. How many people would you need to have in a room before it was more likely than not that two had the same last four digits? How many numbers could be issued before it would be more likely than not that there is a duplicate number? What would the answers for the above questions be if there were 13 digit social security numbers? Try to give exact numerical answers.
2. Suppose each person gets a random hash value from the range $[1 \dots n]$. (For the case of birthdays, n would be 365.) Show that for some constant c_1 , when there are at least $c_1 \sqrt{n}$ people in a room, the probability that no two have the same hash value is at most $1/e$. Similarly, show that for some constant c_2 (and sufficiently large n), when there are at most $c_2 \sqrt{n}$ people in the room, the probability that no two have the same hash value is at least $1/2$. Make these constants as close to optimal as possible.

Hint: you may use the fact that

$$e^{-x} \geq 1 - x$$

and

$$e^{-x-x^2} \leq 1 - x \quad \text{for } x \leq \frac{1}{2}.$$

3. For the document similarity scheme described in class, it would be better to store fewer bytes per document. Here is one way to do this, that uses just 48 bytes per document: take an original sketch of a document, using 84 different permutations. Divide the 84 permutations into 6 groups of 14. Re-hash each group of 14 values to get 6 new 64 bit values. Call this the *super-sketch*. Note that for each of the 6 values in the super-sketch, two documents will agree on a value when they agree on all 14 of the corresponding values in the sketch. Why does it make sense to simply assume that this is the only time a match will occur?

Consider the probability that two documents with resemblance r agree on two or more of the six sketches. Write equations that give this probability and graph the probability as a function of r . Explain and discuss your results.

What happens if instead of using a 64 bit hash value for each group in the supersketch, we only use a 16 bit hash? An 8 bit hash?
4. Prove that 636127 is composite by finding an appropriate witness. Be sure to give ample evidence showing that your witness is in fact witnesses. (Note: do not use a factor as a witness! Sure, these numbers are small enough that you can exhaustively find a factor; that is not the point. A factor is not a witness, according to our definition.) Hint: you will want to write some code. You will preferably use a package that deals with big integers appropriately, as you may want to use some of this code for the next problem (RSA). We don't need a code listing for this problem— a short summary of the output should suffice.

The number 294409 is a Carmichael number. Prove that it is composite by finding a witness. Briefly explain why Fermat's little theorem won't help.
5. My RSA public key is: (46947848749720430529628739081,37267486263679235062064536973). Convert the message

Give me an A

into a number, using ASCII in the natural way. (So for “A b”: in ASCII, A = 65, space = 32, and b = 98; translating each number into 8 bits gives “A b” = 010000010010000001100010 in binary.) Encode the message as though you were sending it to me using my RSA key, and write for me the corresponding encoded message in decimal.