Homework #6

Out : October 14, 2:30 pm 1997
Due : October 21, 12:30 pm 1997

1. (a) For each of the following numbers given in binary, give the decimal representation, showing how you obtained your answer.
   i. 101
   ii. 1111
   iii. 10100111010

   (b) For each of the following numbers given in decimal, give the binary representation, showing how you obtained your answer.
   i. 7
   ii. 16
   iii. 632

2. For each of the following numbers x, estimate $\log_{10} x$. Your answer need not be exact. We ask only that your answer be an integer within 1 of the correct answer. We also request that you not use a calculator for this.
   (a) 865733500
   (b) 3257100392479038
   (c) 2903483223973759397330024
   (d) 44095663833379974090552526356 3

3. For each of the following numbers x, estimate $\log_2 x$. Again, your answer need not be exact. We ask only that your answer be an integer within 1 of the correct answer. We also request that you not use a calculator for this.
   (a) 75777 (decimal)
   (b) 3337150 (decimal)
   (c) 10111011001 (binary)
   (d) 10101010000000111111111010 (binary)

4. In class we showed the operations needed to do modular exponentiation using a table like the one below, which shows how to calculate b^{2184} modulo some unspecified number. You should interpret each multiplication as being done with respect to the unspecified modulus.

NAME	COMMAND	formula for value computed
result0	b	b^1
result1	result0 · result0	b^2
result2	result1 · result1	b^4
	⋮	⋮
result12	result11 · result11	b^{2048}
finalresult	result3 · result7 · result12	b^{2184}

For each of the exponentials below, first obtain the binary expansion for the exponent, showing your work. Notice how many bits are needed in your binary expansion. Next, prepare a table very much like the one above indicating the operations needed to calculate the exponential. Your table should have one row for each bit of the exponent's binary expansion, plus one additional row for multiplying together the appropriate results to obtain the desired final result. Finally, count up the number of multiplications needed by the calculation you propose.

(a) b^{16}

(b) b^{128}

(c) b^{257}

(d) b^{194}

5. You will need a regular (nonMark) calculator for this problem. In this problem, we let two computers, MegaComp and SlowThink, compete head on head. Our "racetrack" is the calculation of $b^{938115730327217}$ modulo 3068862310548665. Note that the exponent and the modulus are 15-digit numbers.

   MegaComp is very fast and very dumb. Its designers didn't know about the repeated squaring method, so it uses the naive method for calculating modular exponentials. (Recall from class the formula for the number of operations required for this algorithm.) However, MegaComp performs a million (1,000,000) steps per second.

   SlowThink is very slow. It was designed in an earlier era, and performs only a thousand steps per second. However, it uses the repeated squaring method for calculating modular exponentials.

   Start your computers at the sound of the gun...

   (a) How long will MegaComp take to complete its calculation of the exponential mentioned above. *Hint: Your answer may be best expressed in years.*

   (b) Give a good overestimate of how long SlowThink will take.

6. You have seen in class how to make a table of modular exponentials, as shown below:

   Mod 7 exponentiation (base 3)

x	3^x
0	1
1	3
2	2
3	6
4	4
5	5

Such a table can be used to determine *modular* logarithms (also known as *discrete* logarithms). For a given number y, mod7log$_3 y$ is the smallest nonnegative number x such that $3^x \equiv y \pmod 7$. For example, the mod7log$_3$ of 6 is 3. (Just find 6 in the right column, and look to its left to find 3.)

For this problem, construct a similar table for a modulus of 11 and a base of 3, and provide it to us. Use it to determine the following modular logarithms.

(a) mod11log$_3$ 6 = ?

(b) mod11log$_3$ 2 = ?

(c) mod11log$_3$ 4 = ?

7. Calculating a modular logarithm is a computational problem:

inputs: m, b, y

output: The smallest nonnegative integer x such that $b^x \equiv y \pmod m$

The previous problem should suggest to you an algorithm for this computational problem, namely build an exponentiation table based on the values of m and b, and then look up y in the second column, and output the corresponding exponent x. For a modulus of m, the table must have $m - 1$ rows. (We will discuss the reason for this later in the term.)

(a) How many multiplications are needed to construct such a table? Give a formula depending on m.

(b) How many ticks are needed? Give a formula overestimating the number of ticks depending on m.

(c) Roughly how long would MegaComp take using this algorithm to calculate a modlog if the modulus is a fifty-digit number?

8. There is a better algorithm known for computing a modlog than the one that builds a table. (It's too complicated to explain in this class.) The number of ticks required when the modulus is m is estimated by the following formula (where k is the number of digits in the modulus):

$$k^2 \cdot 10^{\sqrt{k \cdot (\log_{10} k)}}$$

(a) How long would SlowThink take to compute a modlog using the better algorithm if the modulus m is a fifty-digit number? Use a regular calculator to plug into the above formula.

(b) Now MegaComp has been outfitted with the better algorithm for modlog. Our aim here is to explore its limits. How how many digits long would the modulus have to be in order for MegaComp to take more than a thousand years to calculate a modlog? (Try 50 digits, 60 digits, 70 digits...)