Math 110 Homework 7 Solutions

February 26, 2015

1. (a) Describe the Exponent Factorization Method (Chapter 6.4). What additional information do we need to use this factorization method that is usually prohibitively difficult to obtain?

   (b) Suppose I know that $2^{900} \equiv 1 \pmod{9191}$. Use the Exponent Factorization Method to factor 9191.

   Solution: The Exponent Factorization method tries to factor n by finding a square root of 1 which is not $\pm 1 \pmod{n}$. In this regard it is like parts of the Miller-Rabin primality test. It starts with an a and $r > 0$ such that $a^r \equiv 1 \pmod{n}$. This is often very hard to find. Writing $r = 2^k m$, we can consider $b_0 \equiv a^m \pmod{n}$ and $b_i = b_{i-1}^2 \pmod{n}$. Some $b_i \equiv 1 \pmod{n}$ for $i \leq k$, in which case b_{i-1} will be a square root of 1 that might not be $-1 \pmod{n}$ (provided $i \neq 0$).

   For example, we can factor 9191 by observing that $900 = 4 \cdot 225$ and $b_0 \equiv 2^{225} \equiv 3242 \pmod{9191}$ while $b_1 \equiv b_0^2 \equiv 5251 \pmod{9191}$ which is a non-trivial equare root of 1. Then $\gcd(5251 - 1, 9191) = 7 \cdot 1313$. 1313 is not prime, but has an obvious factor of 101. Thus $9191 = 7 \cdot 13 \cdot 101$.

2. Trappe-Washington Chapter 7 Problem 5(a).

   Solution: Let α be a primitive root and β_1 and β_2 be units modulo p. If $\alpha^r \equiv \beta_1 \pmod{n}$ and $\alpha^s \equiv \beta_2 \pmod{n}$, then $\beta_1\beta_2 \equiv \alpha^{s+r} \pmod{n}$. Taking $r = L_\alpha(\beta_1)$ and $s = L_\alpha(\beta_2)$, by Homework 4 Problem 2, this means that $s + r \equiv L_\alpha(\beta_1\beta_2) \pmod{n-1}$.

3. (a) Describe the Pohlig-Hellman Algorithm for computing discrete logarithms (Chapter 7.2.1). In the notation in the textbook, in lecture we described how to compute x_0. Be sure to complete the description by carefully explaining how we can find x_1, x_2, etc.

   (b) Let $p = 71$. The congruence class 11 (mod 71) is a primitive root. Use the Pohlig-Hellman Algorithm to solve $11^x \equiv 30 \pmod{71}$ for the exponent $x \pmod{70}$.

   Solution: We search for an x for which $\beta \equiv \alpha^x \pmod{p}$ using the notation of the book. Let q^r be the largest power of q dividing $p-1$.. Write $x = x_0 + x_1 q + \ldots + x_{r-1}q^{r-1}$. Suppose we already know $x_0, \ldots x_i$. Then consider $\beta_i \equiv \beta\alpha^{-x_0 - qx_1 \ldots - q^i x_i} \equiv \alpha^{x_{i+1}q^{i+1} + \ldots + x_{r-1}q^{r-1}} \pmod{n}$. We calculate

$$\beta_i^{\frac{p-1}{q^{i+2}}} \equiv \beta_i^{x_{i+1}\frac{p-1}{q} + (p-1)(\ldots)} \pmod{p}.$$

   All the omitted terms are multiples of $p-1$, so by Fermat's little theorem they do not effect the value. Furthermore, the qth power of this is one, so if we run through the q powers of $\alpha^{\frac{p-1}{q}}$ which the are the qth roots of unity, we will find the value of x_{i+1}.

   Then do this to find all of the x_i for each prime power dividing $p-1$ and combine the results using the Chinese remainder theorem modulo $p-1$.

   Remember that this requires $p-1$ to only have small factors so it is feasible to factor it and to run through the q powers.

   Since $70 \equiv 2 \cdot 5 \cdot 7$, we can use the algorithm. Suppose $11^x \equiv 30 \pmod{71}$. First let $q = 2$. Then we consider $\frac{p-1}{q} = 35$, and compute $30^{35} \equiv 1 \pmod{71}$. This means that $x \equiv 0 \pmod{2}$. Likewise for

$q = 5$ we compute $30^{\frac{70}{5}} \equiv 1 \pmod{71}$, so $x \equiv 0 \pmod 5$. Finally for $q = 7$ we compute $30^{10} \equiv 20 \pmod{71}$. Which power of $11^{10} \equiv 32 \pmod{71}$ is it? Well, running through the choices we see that $32^6 \equiv 20 \pmod{71}$, so $x \equiv 6 \pmod 7$. Combining these via the Chinese remainder theorem, we see that $x \equiv 20 \pmod{70}$ is the discrete logarithm.

4. (a) Describe the Baby Step, Giant Step method for computing discrete logarithms (Chapter 7.2.2).
   (b) Using this method with $N = 10$, find all solutions x to $5^x \equiv 2 \pmod{97}$. Note that 5 is a primitive root of the prime 97.

   Solution: The description in the book is fine.

   We compute powers of 5 for the first list:

$$5, 5^2 \equiv 25, 5^3 \equiv 28, 5^4 \equiv 43, 21, 8, 40, 6, 30 \pmod{97}.$$

   For the second list, we compute $\beta\alpha^{-10k} = 25^{-10k}$ for $0 \leq k < 10$ and compare them to the first list. The list begins $2, 22, 48, 43 \ldots$. We stop when we notice that $5^4 \equiv 43 \equiv 25^{-30} \pmod{97}$. Thus the discrete logarithm of 2 is 34.

5. Trappe–Washington Chapter 7 Problem 12.

   Solution: With the notation of the book, think about $j + Nk$ as a two digit number base N. This represents every integer between 0 and $N^2 - 1$. As $m \equiv c^d \pmod n$, and $d < N^2$, d can be written as $d = j + Nk$. Then the elements c^j and mc^{-Nk} agree modulo n, so two elements match. Given the match, you recover d in terms of the j and k used.

   Of course, this may not be the actual decryption exponent, it is simply a number with $c^d \equiv m \pmod n$. For example, if we used $n = 15$ and $e = 3$ and $d = 3$ then the message $m = 4$ encrypts to $c = 4 \pmod{15}$. But then $c^1 = 4$ matches mc^{-0}, so we find $j = 1$ and $k = 0$. This shows that $c^1 \equiv 4 \pmod{15}$ but does not tell us the original decryption exponent.

   This is not efficient though, as the size of the lists needs to be N, where N is around the square root of the modulus n used. Trial division takes at most $\sqrt{n}$ steps, so this is no faster than factoring n through brute force.

6. (a) Describe the Index Calculus method of computing discrete logarithms (Chapter 7.2.3).
   (b) Given that 3 is a primitive root of the prime 101, find all solutions x of $3^x \equiv 96 \pmod{101}$ using the Index Calculus. It may help you to know:

$$\begin{aligned} 3^1 &\equiv 3 \pmod{101} \\ 3^{16} &\equiv 16 \pmod{101} \\ 3^{21} &\equiv 50 \pmod{101} \\ 3^{22} &\equiv 49 \pmod{101} \\ 3^{27} &\equiv 90 \pmod{101} \\ 3^{30} &\equiv 6 \pmod{101} \end{aligned}$$

   Solution: The description in the book is fine.

   The problem gives us a nice set of powers of 3 that are products of the small primes $2, 3, 5, 7$. These in turn give us information about the discrete logarithms of $2, 3, 5, 7$. For example, the first tells us the discrete logarithm of 3 is 1, while combining this with last tells us that

$$3^{30} \equiv 3 \cdot 3^{L_3(2)} \pmod{101}$$

so $L_3(2) \equiv 29 \pmod{100}$. You can get more information using linear algebra. To find the discrete logarithm of 96, we simply compute $96 \cdot 3^x \pmod{101}$ for some values of x. We then see if we can express the result in terms of the factor base. Well, taking $x = 0$ we see that $96 = 2^5 \cdot 3$. But $2^5 \equiv 3^{5 \cdot 29} \pmod{101}$, so $96 \equiv 3^{1+5 \cdot 29} \pmod{101}$. Thus the discrete logarithm is 46 (modulo 100).

7. Let $\mathcal{A}$ be an alphabet of of q symbols (also called letters). In your own words:
   (a) Define a q-ary code of length n.
   (b) Define the *Hamming distance* d on $\mathcal{A}^n$, and explain what it means to say that d is a metric.
   (c) Define the *Hamming sphere*, the closed ball $B(c, r)$ of radius r around the word c in the Hamming distance. Compute the sets $B(0100, 2) \subseteq (\mathbb{Z}/2\mathbb{Z})^4$ and $B(11, 1) \subseteq (\mathbb{Z}/4\mathbb{Z})^2$. Here, 0100 abbreviates the vector (0,1,0,0), and 11 abbreviates (1,1).
   (d) Define the *minimum distance* $d(C)$ of a code C. What does $d(C)$ tell you about the code?
   (e) Define *nearest neighbour decoding.*
   (f) Define an (n, M, d) code.
   (g) Define the *code rate* of a code C. What does the code rate tell you about C?

   Solution: A $q-$ary code of length n is a subset of $\mathcal{A}^n$ (which represents which words of length n are used to encode things).

   The Hamming distance between two elements of $\mathcal{A}^n$ is the number of positions where the elements differ. It being a metric mean it behaves like a generalization of a distance function: you should think that the Hamming distance is small when words are close together, and you can often reason informally about codes by drawing pictures of balls. Formally, being a metric means that

   - $d(u, v) \geq 0$ for any $u, v \in \mathcal{A}^n$, $d(u, v) = 0$ if and only if $u = v$,
   - $d(u, v) = d(v, u)$
   - $d(u, v) \leq d(u, w) + d(w, v)$

   The Hamming sphere $B(c, r)$ is $\{w \in \mathcal{A}^n : d(c, w) \leq r\}$. If $c = 0100$ and $r = 2$, this is all words which agree with c in at least two places. The list is

   $$\{0100, 1100, 0000, 0110, 0101, 1000, 0010, 1110, 1101, 0111, 0001\}.$$

   For a four element alphabet with $c = 11$ and $r = 1$, the list is all things which agree with c in at least one place: ie all code words where the first letter is 1 or the second is 1 (seven of them).

   The minimum distance of a code is the minimum Hamming distance between code words. It tells you how many errors will change a code word into another valid code word.

   The nearest neighbor decoding attempts to correct errors by correcting $w \in \mathcal{A}^n$ to the closest codeword (with respect to the Hamming distance).

   An (n, M, d) code is a code of length n with M code words and minimum distance d. This measures the properties of the code relevant to analyzing its efficiency.

   The code rate is $\frac{\log_q(M)}{n}$. This is the ratio of $\log_q(M)$, the information content of a code word, to the information content of a n-letter word, so this measures the ratio of the data in the message to the data transmitted by the code.

8. Determine (n, M, d) and the code rate R of the following codes:
   (a) Let $\mathcal{A}$ be an alphabet of q letters, and let $C = \mathcal{A}^n$ be the set of all q-ary n-tuples.
   (b) Let $\mathcal{A} = \mathbb{Z}/q\mathbb{Z}$ and let C be the set of all words $(a_1, a_2, \ldots a_n)$ in $\mathcal{A}^n$ such that

   $$a_1 + a_2 + \ldots + a_n \equiv 0 \pmod{q}.$$

(c) Let $\mathcal{A}$ be the alphabet on q elements $\{0, 1, \ldots q-1\}$. Let C be the length-n repetition code

$$\{(0, 0, \ldots, 0), (1, 1, \ldots, 1), \ldots (q-1, q-1, \ldots, q-1)\} \subseteq \mathcal{A}^n.$$

(d) Let $\mathcal{A}$ be the alphabet on q elements $\{0, 1, \ldots q-1\}$, and let C be the set of all words of the form $(a, a, a, \ldots, a, 0, 0, 0, \ldots 0)$ having k copies of $a \in \mathcal{A}$ and $(n-k)$ copies of 0.

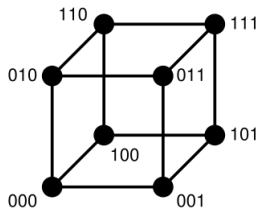
Solution: For the first, the number of code words is q^n, so $\log_q(q^n)/n = 1$ is the rate. The minimum distance is 1.

For the second, there are q^{n-1} code words: pick $n-1$ entries arbitrarily, and use the relation to find the last one. Thus the code rate is $\log_q(q^{n-1})/n = (n-1)/n$. The minimum distance is 2: the linear equation shows code words cannot differ in one spot, and it is easy to find code words that differ by 2.

For the third, there are q code words, so the code rate is $1/n$. The minimum distance is n, because all of the code words differ in n spots.

For the fourth, there are n code words (having 1 to n of a) for each $a \neq 0$. There is also a code word of all 0s. This is $(q-1)n+1$ code words. Thus the rate is $\log_q((q-1)n+1)/n$. The minimum distance is 1, because $1000\ldots$ differs from $0000\ldots$ in one spot.

9. Here is a schematic of $(\mathbb{Z}/2\mathbb{Z})^3$, with a line drawn between all points of Hamming distance 1.



(a) Recopy the picture. Using a different colour, connect all points of Hamming distance 2. Using a third colour, connect points of Hamming distance 3.

(b) Make an analogous schematic of $(\mathbb{Z}/2\mathbb{Z})^4$ and of $(\mathbb{Z}/3\mathbb{Z})^2$, with lines drawn between all points of Hamming distance 1.

(c) Choose a binary length 4 code C in $(\mathbb{Z}/2\mathbb{Z})^4$ of minimum distance $d(C) = 3$ including at least 2 codewords. In your schematic of $(\mathbb{Z}/2\mathbb{Z})^4$, colour each codeword, and draw a Hamming sphere of radius 2 around each codeword. Interpret what "minimum distance 3" means in terms of these Hamming spheres and the geometry of your picture. Explain why your code can detect $s = 2$ errors.

(d) In your diagram of $(\mathbb{Z}/2\mathbb{Z})^4$, draw Hamming sphere of radius 1 around each codeword. Explain why your code can correct $t = 1$ error.

(e) Write a proof (in your own words) of the following result, which appears on page 400.

Theorem 1. *1. A code C can detect up to s errors if $d(C) \geq s+1$.*
2. A code C can correct up to t errors if $d(C) \geq 2t+1$.

Solution:

(a) The points of Hamming distance two are the opposite corners of a face. The points of Hamming distance three are opposite corners of the square.

(b) $(\mathbb{Z}/2\mathbb{Z})^4$ looks like a hypercube with the points of Hamming distance one being the points joined by an edge. $(\mathbb{Z}/3\mathbb{Z})^2$ looks like a three by grid, where things in the same row or column are connected.

(c) For example, take 0000 and 0111 to be the code words. The minimum distance is obviously three, and this means that a ball of radius three around some code word contains the other. Now consider the Hamming balls of radius two. The one around 0000 contains everything with at most two 1, for

example. In particular, it does not contain 0111, and vice versa for the ball of radius two around 0111. This means that the code can detect up to two errors, for no code word is within two of another.

(d) The balls of radius one around each code word are disjoint: the one around 0111 only contains elements with at least two 1s, while the one around 0000 contains at most one 1. Therefore if we introduce at most one error to a code word, we end up in one of the balls. Since they are disjoint, we can tell which code word we started with and correct the error.

(e) If $d(C) \geq s+1$, then starting with a code word and introducing s errors produces an element that is Hamming distance $\leq s$ from a code word. Since the minimum distance between code words is $d(C)$, this new element cannot be a different code word. Thus after introducing errors we do not have a code word, so we can detect errors.

Now suppose $d(C) \geq 2t+1$, and suppose we introduce up to t errors to a code word w to get w'. In other words, $d(w, w') \leq t$. Let v be another code word. Then by the triangle inequality and the inequality $d(C) \geq 2t+1$ we obtain

$$2t+1 \leq d(w,v) \leq d(w,w') + d(w',v).$$

Rearranging we see that $d(w', v) \geq 2t+1-t = t+1$. Therefore there is a unique code word within distance t of w' which allows us to correct the error.

10. Let $\mathcal{A}$ be an alphabet of of q symbols (eg, $\mathcal{A} = \mathbb{Z}/q\mathbb{Z}$).
    (a) How many elements are there in $\mathcal{A}^n$?
    (b) Let w be a fixed word in $\mathcal{A}^n$. How many words in $\mathcal{A}^n$ are Hamming distance 0 from w? How many words in $\mathcal{A}^n$ are Hamming distance exactly 1 from w?
    (c) For $m \geq 0$ in $\mathbb{Z}$, how many words in $\mathcal{A}^n$ are Hamming distance exactly m from w?
    (d) How many words are in the ball $B(w, r)$ for fixed radius $r \geq 0$?
    (e) Prove the Hamming Bound, the theorem on page 404.

If you're not sure how to proceed, start by working out the answer for some small values of q and n. Recall that $\binom{a}{b} = \dfrac{a!}{(b!)(a-b)!}$ is the number of ways to select a subset of b (unordered) elements from a set of a elements.

Solution: There are q^n elements in $\mathcal{A}^n$.

There is one word (w) with Hamming distance 0 from w. There are $(q-1)n$ words that are Hamming distance one: modify one of the positions to any of the $q-1$ other choices.

In general, a word that is Hamming distance m from w arises by picking m spots from the n symbols of w and changing the entry. There are $\binom{n}{m}$ ways to choose m things from n things when the order doesn't matter. For each of these, we can change the symbol to any of the other $q-1$ choices. So there are $\binom{n}{m}(q-1)^m$ words of Hamming distance exactly m from w.

For a ball of radius r, we sum up the words of Hamming distance m for $m = 0, 1, \ldots [r]$:

$$\sum_{0 \leq m \leq r} \binom{n}{m} (q-1)^m$$

For the Hamming bound, the idea is you have non-overlapping spheres around each code word. Since you know how many elements are in each sphere, the number of spheres times the number of the elements of the sphere must be less than the total number of words. There are q^n possible words, the sphere of radius t has

$$\sum_{0 \leq m \leq t} \binom{n}{m} (q-1)^m$$

elements, and there are M code words, giving the stated bound.

11. Read Example 4 in Section 18.1, on the Hamming [7,4] code. Complete Trappe–Washington Chapter 18 Problem 1.

Solution: Using the notation of Example 4, to detect the errors we want to multiply the vectors $(0,1,0,0,1,1,1)$ and $(0,1,0,1,0,1,0)$ by the transpose of H. We obtain $(0,1,0)$ (the sixth row of H) for the first and $(0,0,0)$ for the second. This means there was an error in the sixth entry of the first, and no error in the second. Therefore the correct code words are 0100101 and 0101010. The first four digits give the original messages of 0100 and 0101.