EECS 70 Discrete Mathematics and Probability Theory
Spring 2013 Anant Sahai

HW 7

Due March 11

1. Error-Detecting Codes

In the realm of error-correcting codes, we usually want to recover the original message if we detect any errors, and we want to provide a guarantee of being able to do this even if there are k malicious errors. Suppose that instead we are satisfied with detecting whether there is any error at all and do not care about the original message if we detect any errors. In class you saw that for recovering from at most k malicious errors when transmitting a message of length n you need to extend your message by $2k$ symbols and send a message of length $n+2k$. But since we don't require recovering the original message, it is conceivable that we might need less symbols.

1. Formally suppose that we have a message consisting of n symbols that we want to transmit. We want to be able to detect whether there is any error if we are guaranteed that there can be at most k malicious errors. How should we extend our message (i.e. by how many symbols should we extend, and how should we get those symbols) in order to be able to detect whether our message has been corrupted on its way? You may assume that we work in $GF(p)$ for a very large prime number p.
2. If you were to detect non-malicious errors (i.e. random perturbations of some symbols), how would you do this using just one additional symbol? Obviously you won't be able to guarantee detection, but provide a reason why you think most cases of non-malicious errors will be detected by your algorithm.

2. Spies

The president wants to authorize military generals to launch nuclear weapons without the direct approval of the president if enough of them sanction this launch. As you might remember, secret sharing schemes come in handy here. But now there is a new twist. We have been informed that spies have infiltrated the army, and have even become generals. When it is time for a nuclear launch, the spies can give false information and therefore break the usual secret-sharing scheme.

There are 100 generals in the military. The president knows that out of those 100 at most 5 are spies. He wants a secret sharing mechanism where any group of 9 generals cannot launch the nuclear missiles. However he wants n generals to always be able to launch the nuclear missiles, no matter how many of the 5 spies are there among them. What mechanism should the president use? What is the correct bound n that guarantees any n generals can launch the missiles (even if there are spies among them).

3. Magic!

In this problem we will investigate what happens when in error-correcting codes there are fewer errors than the decoding algorithm is able to handle. For the entire problem we are working in $GF(7)$.

Assume that we wish to transfer a message of length 2 which we denote by (m_1, m_2). Each m_i is a member of $GF(7)$. We also wish to be able to correct up to $k = 2$ errors. Using the error-correcting codes we learned in class, we have to first find a polynomial $P(x)$ of degree at most 1 such that $P(1) = m_1$ and $P(2) = m_2$. Then we have to extend the message we send by $2k$ symbols. i.e. we will send $P(1), P(2), P(3), P(4), P(5), P(6)$ to the recipient.

1. Consider an example where $(m_1, m_2) = (4, 2)$. What are the six symbols that are transmitted?
2. Now assume that you have received these numbers: $5, 3, 4, 0, 3, 6$. i.e. if there were no errors then we would have $P(1) = 5, P(2) = 3, P(3) = 4, P(4) = 0, P(5) = 3, P(6) = 6$. Now, write down the linear equations that help decode error-correcting codes.
3. In this part try to solve the linear equations you got in the previous section. You should observe that there are multiple solutions to these equations. Pick two different solutions and for each one write down the error-locating polynomial $E(x)$ and the polynomial $Q(x)$. In each of the two solutions divide $Q(x)$ by $E(x)$ to get the original polynomial. Do you get the same polynomial in both cases?
4. Factorize $E(x)$ in each one of the two solutions you got to get its roots. Do they have a common root? What does that tell you about the position of errors in the transmitted message?

4. Graphs

A graph G is called bipartite if its vertices can be divided into two disjoint set L, R such that edges only connect one vertex in L to one vertex in R. There is no edge that connects two vertices in L or two vertices in R.

1. Prove that G has no cycles of odd lengths.
2. Prove that $\sum_{v \in L} degree(v) = \sum_{v \in R} degree(v)$.

5. Count it

1. Suppose that a pizzeria lets you choose toppings for the pizza among 10 different ingredients. You have budget for buying a pizza with 3 toppings. How many pizzas can you order with 3 different toppings?
2. What if you are allowed to have repeated toppings in the previous question?
3. Consider a standard deck of 52 cards, consisting of four suits each containing thirteen cards labelled $A, 2, \ldots, 10, J, Q, K$. You receive a hand of four cards. How many possible 4-card hands are there? [Note: The order of the cards is not important.]
4. How many 4-card hands consist only of "face cards" (i.e., A, J, Q, K)?
5. How many 4-card hands contain at least one Queen?
6. How many 4-card hands contain two cards of one suit and two of a different suit?
7. How many 4-card hands contain four different suits?
8. How many 4-card hands contain a contiguous sequence of four values (e.g., $4, 5, 6, 7$ or $A, 2, 3, 4$), regardless of the suits? [Note: Ace may be treated both as a high card and a low card, so we include both the sequence J, Q, K, A and $A, 2, 3, 4$.]
9. How many different anagrams (i.e. permutations of letters) of the word piazzza (with three z's) are there?
10. How many binary (i.e. 0, 1) strings of length 10 are there that have four 1's?
11. We have n boxes of different sizes, each one big enough to contain all of the smaller boxes at once (when put together side-by-side). We may nest the boxes however we want (i.e. a box put in a box which itself is put in another box is a valid nesting). In the end, all boxes should be directly or indirectly nested in the largest box. In how many ways can we nest the boxes?

6. Prove it

Prove the following statements by a combinatorial argument without doing any explicit calculations (i.e. tell us a story about two different ways of counting something, and tell us why the two ways result in the two sides of the equality statement).

1. $\binom{n}{i}\binom{n-i}{k-i} = \binom{k}{i}\binom{n}{k}, \quad 0 \le i \le k \le n$
2. $\binom{m+p}{n} = \sum_{k=0}^{m} \binom{m}{k}\binom{p}{n-k}, \quad m+p \ge n, \;\; m,n,p \ge 0$

7. Fibonacci and combinatorics

Fibonacci numbers are defined recursively in the following way:

$$F(n) = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n \ge 2 \end{cases}$$

Prove that $F(n) = \sum_{k=0}^{n} \binom{n-k}{k}$. [Hint: Use the property $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.]