EECS 121 Coding for Digital Communication and Beyond
Fall 2013 Anant Sahai MT 2

PRINT your student ID: ____
PRINT AND SIGN your name:

__________________, __________________ ______________________
(last) (first) (signature)

PRINT your Unix account login: ee121-_____________

Prob. 1	
Prob. 2	
Total	

You may consult your one handwritten note sheet. **(You must turn it in with your exam.)** Phones, calculators, tablets, and computers are not permitted for the in-class phase. No collaboration is allowed at all and you are not allowed to look at another's work or even talk/text/im/etc. to a fellow 121 student until you've both completed the take-home part of the exam. You are also not allowed to get any help from other people beyond the Prof and GSI.

Please write your answers in the spaces provided in the test; in particular, we will not grade anything on another exam page unless we are clearly told in the problem space to look there.

You have 80 minutes in-class. There are 2 questions, of varying numbers of points. The questions and their parts are of varying difficulty, so avoid spending too long on any one part during the in-class phase.

During the take-home phase, the main thing is for you to do the computer implementation of the parts that require it as well as testing those parts. However, you may also correct any errors in the non-computer parts to recover a fraction of the points that you might have missed. You are not allowed to use any code or algorithms, etc. found online to help you. You are, however, permitted to use the code from your own course projects and hws to build upon. But you must do it yourselves.

60 Points is a good score.

Do not turn this page until your instructor tells you to do so.

Problem 1. Gotta catch them all (30 points)

This question is related to the final spike in the robust Soliton distribution.

There are k data symbols (not bits, think of them as being in a very large finite field), all of which are unknown.

a. (5 points) Suppose that we have already decoded $k-L$ data symbols correctly. So L symbols remain.

A new coded symbol arrives that is the sum of d data symbols. These d symbols are *each* drawn uniformly at random from the k potential symbol positions. (*NOTE: this is different from the regular robust soliton distribution*)

What is the probability that this coded symbol is totally useless at this point because all of the data symbols contained within it have already been decoded?

b. (5 points) Continuing the previous part, what is the probability that this symbol can be used immediately to decode one new data symbol?

c. (5 points) Consider $L = R$. Assume k is very large, $d = \frac{k}{R}$, and R is significantly smaller than k. Approximately evaluate the probability in the previous two parts.

d. (5 points) Continuing the previous part. Assume k is very large, $d = \frac{k}{R}$, and R is significantly smaller than k. Now consider general $1 \leq L < R$. Approximately evaluate the probability of a new coded symbol being able to immediately decode one new data symbol.

e. (10 points) Use the previous result to calculate the approximate (up to scaling $O(\cdots)$) expected time to complete decoding if we continue to receive coded symbols with $d = \frac{k}{R}$ assuming that we start with exactly $L = R$ symbols left undecoded and get one new iid coded symbol per unit time.

Assume that if the symbol isn't immediately useable to decode one new data symbol, it is discarded.

(HINT: this turns out to be like the coupon-collector problem.)

extra page

Problem 2. Two pipes being coded together (60 points)

In this problem, we have three different channels:

- Red (unevolved) channels are quite noisy and have a quality parameter q.
- Yellow (half-evolved) channels are a less noisy and have a quality parameter $p < q$.
- Green (fully evolved) channels are perfectly reliable and carry bits perfectly.

We are going to use the codes of Homework 3 to protect the bits, however the encoder doesn't discriminate between the bits in the yellow and red channels. Everything is coded together. To be precise, we have $\frac{k}{2}$ uncoded data bits that are sent over the yellow channel and received unreliably. A further $\frac{k}{2}$ uncoded data bits are sent over the red channel and received even more unreliably. The code simply takes a binary matrix H and hits the k bits to get s parities which are transmitted on the green channel where they are received perfectly.

Like the HW, consider H to be filled with iid Bernoulli-$\frac{1}{2}$ entries. It is known to the decoder.

The decoder knows both p and q as well as which bits were sent on the yellow channel and which were sent on the red channel.

a. (5 points) Consider an erasure-type channel with $q > p$ being probabilities of bit erasure. Assume k is large and s is proportional to k.

How big of an s do you need so that you can most likely use the green channel information to be able to reliably (with very low probability of failing) recover the original data sent in the red and yellow channels.

PRINT your name and student ID: ________________________________

b. (10 points) Repeat the previous part for an error-type channel where p and q represent the probabilities of iid flipping the bits in their respective channels. Assume here $\frac{1}{2} > q > p$.

How big of an s do you need?

Argue why this s is large enough. (A fully formal proof is not needed.)

(HINT: How many flips do you expect to see in the yellow channel? How many in the red channel? What is the probability that there will be significantly more or fewer flips in either channel? About how many different ways are there to get that pattern of flips?

c. (10 points) Give an explicit syndrome-style decoder for the previous part.

(HINT: write out the "type" of the noise sequence as a pair i, j where i represents the number of flips in the red channel and j represents the number of flips in the yellow channel. Notice that each noise sequence has a probability that depends only on its type. List the types in descending order of the probability of those individual noise strings. Now, can you get the syndrome of the noise sequence from the received information? Which noise sequence would you guess given that you know its syndrome?)

d. (35 points) How would you implement the syndrome-style decoder for this problem assuming that s and k were moderate values and p and q were known. (Do this part in-class)

Implement your syndrome style decoder and test it out. (describe your testing strategy in class)

Consider $p = 0.01$ and $q = 0.1$. Consider $k = 20$ and $s = 10$. Use a computer to calculate (or estimate) the probability of error in using your syndrome-style decoder. (describe the approach in class)

[Extra Page]