EE376A - Information Theory
Midterm, Tuesday February 9th

Instructions:

- You have **two hours**, 6:30PM - 8:30PM
- The exam has 2 questions, totaling 100 points. There is an extra credit question for additional 20 points.
- Please start answering each question on a new page of the answer booklet.
- You are allowed to carry the textbook, your own notes and other course related material with you. Electronic reading devices [including kindles, laptops, ipads, etc.] are allowed, provided they are used solely for reading pdf files already stored on them and not for any other form of communication or information retrieval.
- You are required to provide a detailed explanation of how you arrived at your answers.
- You can use previous parts of a problem even if you did not solve them.
- As throughout the course, entropy (H) and Mutual Information (I) are specified in bits.
- log is taken in base 2.
- Throughout the exam 'prefix code' refers to a variable length code satisfying the prefix condition.
- Good Luck!

1. Coin Toss Experiment and Golomb Codes *(50 points)*

Kedar, Mikel and Naroa have been instructed to record the outcomes of a coin toss experiment. Consider the coin toss experiment $X_1, X_2, X_3, ...$ where X_i are i.i.d. $Bern(p)$ (probability of a H (head) is p), $p = 15/16$.

(a) (*5 points*) Kedar decides to use Huffman coding to represent the outcome of each coin toss separately. What is the resulting scheme? What compression rate does it achieve?

(b) (*10 points*) Mikel suggests he can do a better job by applying Huffman coding on blocks of r tosses. Will his scheme approach the optimum expected number of bits per description of source symbol (coin toss outcome) with increasing r? How does the space required to represent the codebook increase as we increase r?

(c) (*20 points*) Naroa suggests that, as the occurrence of T is so rare, we should just record the number of tosses it takes for each T to occur.
To be precise, if Y_k represents the number of trials until the k^{th} T occurred (inclusive), then Naroa records the sequence:

$$Z_k = Y_k - Y_{k-1}, \; k \geq 1, \tag{1}$$

where $Y_0 = 0$

i. What is the distribution of Z_k, i.e., $P(Z_k = j), j \in 1, 2, 3, ...$?
ii. Compute the entropy and expectation of Z_k.
iii. How does the ratio between the entropy and the expectation of Z_k compare to the entropy of X_k? Give an operational interpretation.

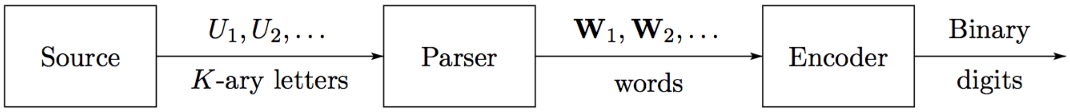
(d) (*15 points*) Consider the following scheme for encoding Z_k, which is a specific case of Golomb Coding. We are showing the first 10 codewords.

Z	Quotient	Remainder	Code
1	0	1	1 01
2	0	2	1 10
3	0	3	1 11
4	1	0	0 1 00
5	1	1	0 1 01
6	1	2	0 1 10
7	1	3	0 1 11
8	2	0	00 1 00
9	2	1	00 1 01
10	2	2	00 1 10

i. Can you guess the general coding scheme [Hint: We are considering quotient and remainder of Z with respect to 4 in this case, and concatenating them in a specific format to form the code for Z]?
ii. What is the decoding rule for this Golomb code?
iii. How can you efficiently encode and decode with this codebook?

2. Tunstall Codes *(50 points)*

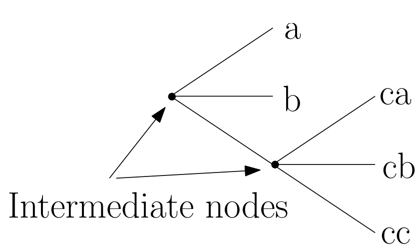
Consider an approach for compression of a memoryless K-ary source $U_1, U_2, \ldots$ distributed according to the discrete random variable U, with $|\mathcal{U}| = K$, illustrated in the following figure:



Instead of encoding the source symbols directly, we first parse them into words with the help of a dictionary of size M. A dictionary is a prefix-free collection of words $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$, where each $\boldsymbol{w}_m$ is a sequence of source letters, with the property that any stream of source symbols can be constructed as a concatenation of words from the dictionary. Each dictionary entry is then represented by a binary codeword of the same constant length b. These binary representations of the dictionary words that make up the source stream comprise the encoder output.

Given a dictionary with words $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$, the parser picks the longest prefix of the source sequence $U_1, U_2, \ldots$ that is in the dictionary (say, $U_1, \ldots, U_L$), and then continues in the same fashion with $U_{L+1}, U_{L+2}, \ldots$, etc.

Consider, for example, $\mathcal{U} = \{a, b, c\}$, and $\boldsymbol{w}_1 =$"a", $\boldsymbol{w}_2 =$"b", $\boldsymbol{w}_3 =$"ca", $\boldsymbol{w}_4 =$"cb", $\boldsymbol{w}_5 =$"cc". The tree associated with this dictionary is given by



(a) (*10 points*) Given a dictionary represented by binary codewords of fixed length b, what should the words of the dictionary satisfy such that the overall compression scheme is optimal (i.e., it achieves the entropy)? Specifically, what should M satisfy, and what should be the probabilities of the words that comprise the dictionary?

Consider the following Tunstall code, for $\mathcal{U} = \{a, b, c\}$, and $P_U(a) = 0.6$, $P_U(b) = 0.3$, and $P_U(c) = 0.1$:

Word	Codeword
b	000
c	001
ab	010
ac	011
aaa	100
aab	101
aac	110

(b) (*6 points*) Compute the entropy of the source $H(U)$.

(c) (*6 points*) Compute the entropy of the dictionary words, denoted by $H(W)$.

(d) (*8 points*) Compute the expected length of the dictionary words, denoted by $E[L_W]$.

(e) (*8 points*) What is the overall compression rate, i.e., the average number of bits of description per source symbol under this compression scheme? Compare this with the compression achieved if the dictionary words were given by "a", "'b" and "c", and the associated binary codewords were of length 2. Which scheme is better?

(f) (*12 points*) Given the dictionary in the above table, can you come up with a better assignment of codewords (not necessarily of the same length) for the same dictionary? The code should be prefix-free. Please construct the best such code. What is the overall compression achieved, i.e., the expected codeword length per source symbol in this case? How does it compare to that of the Tunstall code above? How does it compare to the entropy of the source? Explain.

3. Mix of Problems (Extra credit: *20 points)*

(a) Consider a random variable X supported on $\{1, 2, 3, ...\}$ with $E[X] = 2$.

i. (*5 points*) Show that $H(X) \leq 2$.

ii. (*5 points*) What distribution of the random variable achieves $H(X) = 2$?

(b) Let $X_1, X_2, \ldots$ be an i.i.d. sequence of discrete random variables with entropy $H(X)$. Let

$$C_n(t) = \{x^n \in \mathcal{X}^n : p(x^n) \geq 2^{-nt}\}$$

denote the subset of n-sequences with probabilities $\geq 2^{-nt}$.

i. (*5 points*) Show that $|C_n(t)| \leq 2^{nt}$.

ii. (*5 points*) What is $\lim_{n\to\infty} P(X^n \in C_n(t))$? (distinguish between different values of t)