CSE 21 Spring 2022
Homework 7
Due date: Wednesday, June 1, 2022 at 11:59pm

Instructions

This homework should be done in **groups** of one to four students, without assistance from anyone besides the instructional staff and your group members. Homework must be submitted through Gradescope by a **single representative** of your group and received by **11:59pm** on the due date. There are no exceptions to this rule.

You will be able to look at your scanned work before submitting it. You must **type** your solutions. (hand-drawn diagrams are okay.) Your group representative can resubmit your assignment as many times as you like before the deadline. Only the most recent submission will be graded.

Students should consult their textbook, class notes, lecture slides, podcasts, group members, instructors, TAs, and tutors when they need help with homework. You may ask questions about the homework in office hours, but questions on Piazza should be private, visible only to instructors.
This assignment will be graded for not only the *correctness* of your answers, but on your ability to present your ideas clearly and logically. You should explain or justify, present clearly how you arrived at your conclusions and justify the correctness of your answers with mathematically sound reasoning (unless explicitly told not to). Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to **convince the reader** that your results and methods are sound.
Key Concepts Encoding, Huffman Tree, Ranking

---

1. (20 points)

   Suppose you are an experimental composer and you want to make a melody using 12 notes

   $$(A, A\#, B, C, C\#, D, D\#, E, F, F\#, G, G\#)$$

   Each melody is a sequence of 100 notes with the following frequencies:

A	A #	B	C	C#	D	D#	E	F	F#	G	G#
5	1	16	3	2	13	5	7	9	17	14	8

   (a) Using a *fixed-length* character by character encoding, what is the minimum number of bits needed for each note and how many bits are needed for each melody using this encoding?

   (b) i. Draw the huffman tree for the set of frequencies. (you do not have to show intermediate steps.)

   ii. Give the code for each note. (no justification necessary.)

   iii. Calculate the number of bits needed to encode each melody using the huffman encoding. (show how you calculated this.)

   (c) What is the theoretical minimum number of bits needed to encode each melody? (explain how to do this.)

   (You may need a powerful calculator to do this: https://www.wolframalpha.com/)

2. (12 points) A *slow-growing sequence* of length n is a *non-decreasing* sequence of integers that start with 1 and each pair of entries differ by at most 1.

   Here are a few examples of *slow-growing sequences* of length 8

   $(1, 1, 1, 1, 1, 1, 1, 1)$, $(1, 2, 2, 2, 3, 3, 4, 5)$, $(1, 2, 3, 4, 5, 6, 7, 8)$

   (a) (For $n \geq 1$), How many *slow-growing sequence* of length n are there? (give a brief justification.)

   (b) How many bits would the most efficient encoding of such sequences use? (in terms of n. Justify your work)

   (c) Develop your own encoding/decoding algorithm where the code uses this number of bits. (Please give a brief description of how it works on an arbitrary input.)

   (d) Use your encoding to encode the following slow-growing sequences: (you are not required to provide justification.)

   - $(1, 2, 3, 3, 3, 4, 4, 5)$
   - $(1, 1, 1, 2, 2, 3, 4, 4)$
   - $(1, 2, 2, 3, 3, 4, 5, 6)$

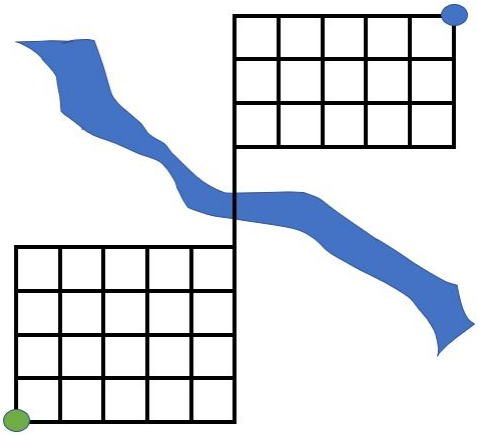
   (e) Use your decoding to decode the following strings: (put "not decodable" if you can't decode the string and give a reason why it is not decodable. Otherwise, you are not required to provide justification.)

   - 01010100
   - 11100011
   - 10111110

3. (20 points)

   There is a town with a river running through it. You live on the south bank at the southwest corner of a 5×6 grid of city blocks. There is a bridge attaching the northeast corner of this grid to the north bank of the town which is comprised of a 3×5 grid of city blocks.

   You wish to travel from the southwest corner of the south bank to the northeast corner of the north bank by only using north or east movements (along with going across the bridge.)

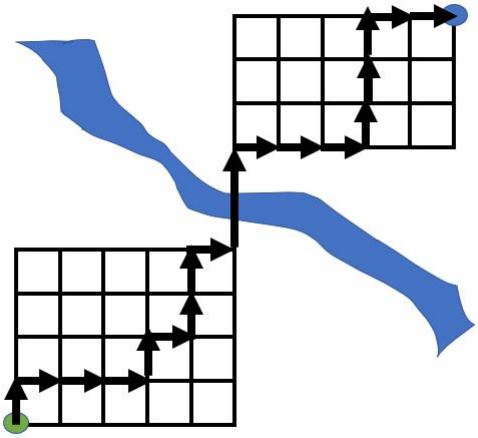


(a) Consider the following encoding of a path:

- Step 1: Convert the path into a string of N's and E's.
- Step 2: Change each N to a 0 and each E to a 1.

i. How many bits does this encoding require?

ii. How would you encode the path: (no justification necessary.)



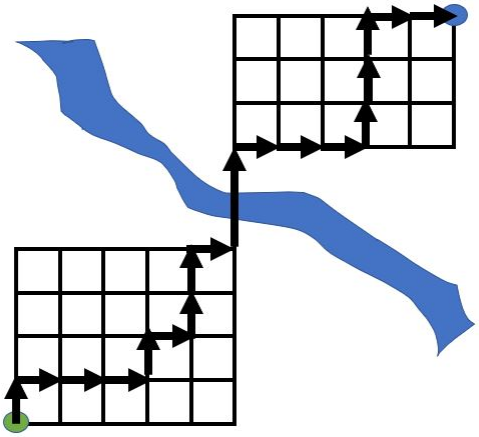
iii. How would you decode the binary string: (you can either draw the path or list the path as a sequence of N and E. Say "non-decodable" if it is not possible to decode the string.)

0 0 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0

(b) What is the theoretically optimal number of bits needed to uniquely encode each path?

(c) Design an encoding algorithm that uses the theoretically optimal number of bits.

(d) Use your encoding algorithm to encode:



(please show your work.)

(e) Based on your encoding algorithm, decode the string: (or put "non-decodable" if it is not decodable.)

0 0 1 1 0 1 1 0 0 1 1 0 0

4. (25 points) Consider the set of all n-character strings over the alphabet: $\Sigma = \{A, S, D, F, G, H, J, K, L\}$. (The second row of a standard keyboard.)

   (a) Using fixed length character by character encoding, how many bits are required to encode each n-character string over the alphabet Σ (in terms of n?)

   (b) Develop a fixed length encoding for this problem.

   (c) Use your fixed length encoding/decoding for the following strings.

      i. Encode the string: "$FLAGS$"
      ii. Encode the string: "$SALAD$"
      iii. Decode the string (or say non-decodable if you cannot decode it): "0000 0001 0010 0011 0100"
      iv. Decode the string (or say non-decodable if you cannot decode it): "1000 1001 1010 1011 1100"

   (d) Determine the number of bits required for the theoretically optimal encoding of each n-character string over the alphabet Σ.

   (e) Develop an encoding algorithm for all 5-character strings over the alphabet Σ that achieves the theoretically optimal number of bits.

      i. Encode the string: "$FLAGS$"
      ii. Encode the string: "$SALAD$"
      iii. Decode the string (or say non-decodable if you cannot decode it): "0000001001000110"
      iv. Decode the string (or say non-decodable if you cannot decode it): "1000100110101011"