1) Take the text otoos1 and do a Maximum Likelihood Estimate of the distribution of characters in the text. Compute the entropy of this distribution. Do the same for the word distribution.

2) Take the second text, otoos6, and:
   i) Compute the Kullback-Leibler divergence from the character distribution in Text 1 to that in Text 2, and vice versa.
   ii) Compute the KL-divergence for the word distributions. Here you will need to smooth the second distribution. One simple method is Laplace (add-1) smoothing: increment all counts by 1 (i.e. replace all zero counts of occurrences of words from Text 1 with 1, and add 1 to non-zero counts).
   iii) Did you need smoothing for the character distributions? Why/why not? Compare the KL-divergences you obtained with regard to the basic unit of observation (letter/word). How do they differ and why?

3) We want to create a simple code for our texts.
   i) Write a procedure that computes a prefix code according to the following algorithm:
      **Input:** An input probability distribution P for a set of symbols S
      **Output:** A binary tree encoding P
      For each symbol \( s \in S \), create a tree leaf node holding \( (s, P(s)) \)
      Place all nodes into a queue Q and order them by probabilities
      while Q contains more than one element do
        Remove the two nodes \( s_1, s_2 \) with the smallest probabilities from Q
        Create a new node with \( s_1, s_2 \) as children and \( P(s_1) + P(s_2) \) as probability.
        Add the new node to the queue
      end while
      return The code tree \( T \), its root is the last node left in Q
      This technique is known as Huffman coding. You can find more details in the Wikipedia article on Huffman coding.
   ii) Write a procedure which computes the code words for all encoded symbols. You can read off the code word for each symbol by traversing the constructed tree. For this, start at the root of the tree and record a 0 for every left branch and a 1 for every right branch you take on the way to the leaf containing the symbol.
   iii) Use the Maximum Likelihood estimate you obtained in 1) from Text 1. Use the obtained code to encode the text. How long is the coded text? What length is theoretically expected, based on the formula from the lecture’s slides? Why the difference?

Your submission should include a brief report on your solution, as well as the source code you wrote to solve the exercise. If possible, avoid using nonstandard libraries not included in the distribution of the programming language you used. Make sure to include brief instruction on how to run your code.

Send your solutions to arif.khan@lsv.uni-saarland.de (if you are attending the Wednesday tutorial) or andrea.fischer@lsv.uni-saarland.de (if you are attending the Tuesday tutorial) by Friday, 24 May.

Important: Please use PDF as a document format. If you need to compress files, use ZIP or GZIP.