1. [15 points] Given an alphabet \( A = (a_1, a_2, a_3, a_4) \) with probabilities
   \[ p(a_1) = 0.2, \quad p(a_2) = 0.3, \quad p(a_3) = 0.15, \quad p(a_4) = 0.35 \]
   find a Huffman code using the first procedure presented in the class and then a
   Huffman code with minimum variance. What is the practical significance of
   finding a minimum variance code?

2. [25 points] Given all the probability of symbols in the alphabet, if every pair of
   symbols are combined to be one and form a new alphabet, will the compression
   ratio be better or worse? Justify it. Compare the n-gram based Huffman coding (if
   n-symbols are represented by one) with single-symbol probability based Huffman
   coding.

3. [30 points] Construct a canonical Huffman code for the following lengths of code
   for an alphabet of 26 symbols:
   2 (letters q and z) codes of length 7, 3 (letters j,k,x) of codes of length 6, 14
   (letters v,g,w,b,y,p,f,m,u,c,l,d,s,h) codes of length 5, 6 (letters i,r,o,a,t) codes
   for the length of 4 and 1 (letter e) code of length 3. Give a code for a message
   ‘compression’ and show how the decoding operation will proceed. Then take a bit
   string ‘001101101011’. Will this decode to a valid string?

4. [30 points] Derive the time and storage complexity of encoding and decoding of a
   minimum variance Huffman code given \( n \) probability values. Do the same for
   canonical Huffman code given \( n \) length values. Be precise in writing the
   algorithms and deriving the complexity results.