I. Huffman codes [1]

Algorithm 1 Huffman(C)

1: \( n = |C| \) \( \triangleright \) Each node \( c \in C \) has a property \( c.freq \)
2: \( Q = C \) \( \triangleright \) \( Q \) is a min priority queue
3: for \( i = 1 \) to \( n - 1 \) do
4: allocate new node \( z \)
5: \( z.left = x = \text{Extract-Min}(Q) \)
6: \( z.right = y = \text{Extract-Min}(Q) \)
7: \( z.freq = x.freq + y.freq \)
8: \( \text{Insert}(Q, z) \)
9: end for
10: return \( \text{Extract-Min}(Q) \) \( \triangleright \) Returns the root and the Huffman Code tree
11: \( \triangleright \) By convention, a 0 is assigned when we take a left branch, while a 1 is assigned when we take a right branch

1) (10 pts) Given the following expected frequencies of letters, form the correct Huffman code:
   \( C = \{ (\text{blank}) : 26, e : 29, a : 18, s : 17, g : 6, m : 4 \} \).

2) (5 pts) Using your code from 1, decode a message: 000110001110110100100111

References