Lab1: HUFFMAN CODING

Through these lab exercises, we will learn how to use the Huffman coding for text compression.

The huffman coding has two parts: encoding and decoding. In the first part of the lab you will learn how to do Encoding and in the second part, you'll learn the decoding.

We provide a basic code for Huffman coding in Python and during the exercises you will ask to change some parts of the code. The basic code for doing the huffman coding is provided *here.*

The **input** for this code is a **string** and the output is the reconstructed formated of the string. The input string can have different items including characters, numbers, space and symbols.

Your code need to be work perfectly with any input string that we give it to.

Part A - Encoding

In this exercise we get the input string which is given in the code and return the encoded (Compressed) format for the input.

- ### Exercise 1
  Compute the frequency of each character in the input string.
- ### Exercise 2
  Sort the frequency table based on occurrence (this will also convert the dict to a list of tuples)
- ### Exercise 3
  Compute the huffman codes for the character of the input string by using the huffman tree.

Part B - Decoding

In this exercise we get the encoded (Compressed) format for the input string and will return the decoded (Decompressed) string.

- ### Exercise 4
  You have to write the decompression function based on the compressed bits and the huffman code for the input string.
- ### Exercise 5
  This part is not related to write the code!

Compute the size that the input string needs and also the size that compressed string needs in bits and compare them with each other and explain your evaluation.