Problem 3: Huffman coding

In this problem, the given string is broken down into unique characters corresponding to their frequencies. After that all unique characters are stored in a min type priority queue. We pick 2 nodes off the priority queue and merge those two nodes and create a new node. The new node's left and right child are the original two nodes. After this we push the new node back into priority queue. We continue this process until the length of priority queue is 1. To assign unique binary codes to all characters we call a recursive function which will traverse the full tree and assign each character a unique binary code. To decode the given string, we move left if we have a '0' or we move right, as soon as we reach a leaf, we add that leaf character to the string and then we start again from root, until we reach end of encoded string.

Time Complexity:

Tree construction: O (d + log(d)) – where d is number of unique characters in the given string

Tree traversal for encoding: O(d) – where d is the number of unique characters

Decoding: O(n): where n is the length of encoded string

Space Complexity:

O(n) – since we are storing a tree full of nodes with characters and frequency