Problem 3: Huffman Coding

Explanation

Design Decisions:

I primarily used Dictionaries to store data as it allows for an $O(1)$ runtime.
The Priority Queue, along with Node having a frequency parameter, allowed me to keep track of the frequencies of each letter. This helped in creating the Tree when having the pop the least frequent letter.
I decided to store the binary codes in the Tree itself so I didn't need to return anything in Tree.create_binary_codes.

Time Complexity:

$O(n)$ since I constantly run through the initial data when determining frequencies, building the priority queue, building the tree, creating the compressed data, and decompressing the data.

Space Complexity:

$O(n)$ from storing the binary codes, priority queue, frequencies.

The rest of the data stored is $O(1)$.