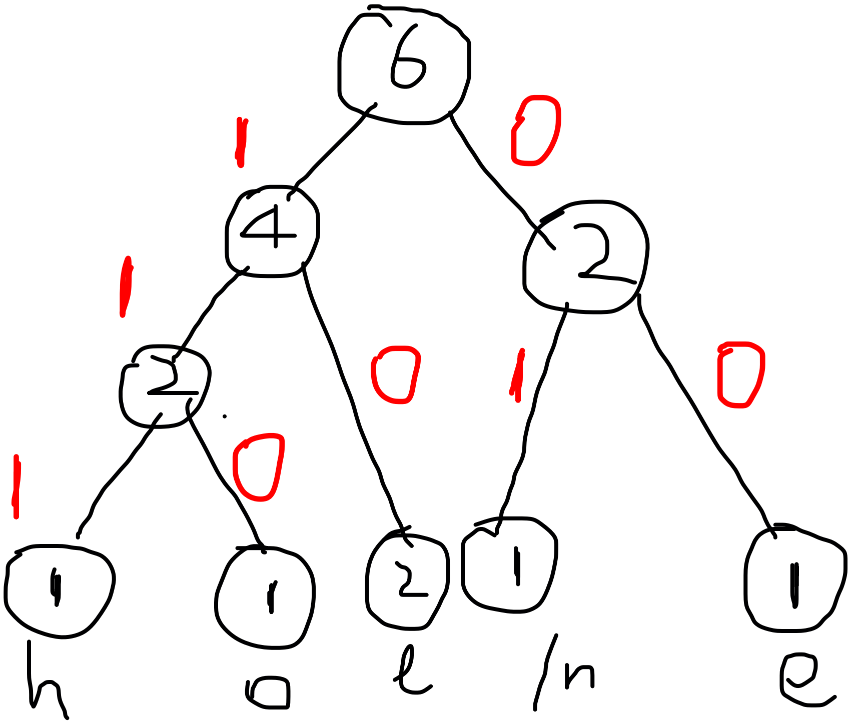
/**
 * compress.pdf for PA3
 * Author: Mingxuan Wang(A53077257) & Jingjue Wang(A97007709)
*/

First, create a priority queue, PQ. Then, get the frequency of each symbol in the input .txt file. And create HCNodes with their frequency and symbol as leaves. Push these HCNodes into the PQ. In PQ, the HCNode with smaller frequency has higher priority. While, the HCNode with larger frequency hs lower priority. If there are two HCNodes with identical frequency, the smaller symbol has higher priority. The HCNodes in in PQ are the leaves in the Huffman Tree. In order to build the tree up, the two nodes with highest priority merge together to create a new node, which is the parent of the two children nodes. The frequency of the parent is the sum of the frequency of the two children. After merging the two leaves, they are popped out from PQ. The parent created is the new element in PQ instead. Then, encode the root node as 0. From the root node, the left child of each node is encoded as 1. And the right child of each node is encoded as 0. And the compressor output is the code without the 0 of the root.
For example, string "hello/n" has a Huffman tree below:

h =111
e = 00
l = 10
o = 110
/n = 01

h e l l o /n
111 00 10 10 110 01



Compared the hand-write code output with the compressor output, they are identical except for the header part in compressor output.