Input: Set of pairs $S = (c_1, f_1), (c_2, f_2), ..., (c_n, f_n)$, where c_i is a character, and f_i is the frequency of the character and $\sum_{i=1}^{n} f_i = 1$
Output: Tree T corresponding to an optimal prefix tree for the specified character pairs.

procedure HUFFMAN_CODE(S)
$Q \leftarrow$ initialize empty priority queue with frequency as key and tree node as value

for $s \in S$ **do**
$node \leftarrow$ create tree node with no children
$node.frequency \leftarrow s.frequency$
$node.character \leftarrow s.character$
Add $node$ to Q

while $|Q| \geq 2$ **do**
$p_1 \leftarrow Q.remove\_min()$
$p_2 \leftarrow Q.remove\_min()$

$node \leftarrow$ create tree node with p_1 as left child and p_2 as right child
$node.frequency \leftarrow p_1.frequency + p_2.frequency$

Add $node$ to Q

if $|Q| = 1$ **then** ▷ When merging the last 2 nodes, we will get one final root node
return $Q.remove\_min()$
else
return UNDEFINED