Create Huffman code tree:

1. Create a leaf node for each symbol and add it to the priority queue.
2. While there is more than one node in the queue:
    1. Remove the two nodes of highest priority (lowest probability) from the queue
    2. Create a new internal node with these two nodes as children and with probability equal to the sum of the two nodes' probabilities.
    3. Add the new node to the queue.
3. The remaining node is the root node and the tree is complete.

O(n log n) time

The Huffman tree yields a code for each symbol. The function below shows how the tree is traversed to obtain the codes from the tree – initial call is print_codes(root, “”)

```
function print_codes(tree, prefix):
    If tree is a leaf:
        Print tree.value (frequency = tree.frequency) maps to code: prefix
    Else
        print_codes(tree.left,  prefix + “0” )
        print_codes(tree.right,  prefix + “1” )
```