Huffman Trees

We want to compress text strings consisting of the symbols `A`, `B`, `C` and `D` using Huffman coding. The relative frequencies are given by the following frequency table:

A	B	C	D
1	3	7	2

(a) What are the code lengths for the different symbols in the corresponding Huffman tree?

(b) What do the leaf nodes in the resulting tree contain?

Generally, in a Huffman tree, each node v has:

- v.`symbol` – The symbol of the node (or **`null`**)
- v.`freq` – The frequency of the node
- v.`left` – Left child of the node (or **`null`**)
- v.`right` – Right child of the node (or **`null`**)

A Huffman tree is built from a set of pairs (s, f) where s is a symbol and f is a frequency.

(c) Describe the algorithm for building a Huffman tree. You can outline it with pseudocode, but a good description in natural language works just as well.