Given a set of symbols and their probabilities, Huffman's Algorithm tells us how to construct an optimal variable-length encoding.

By "optimal" we mean that, assuming we're encoding each symbol one-at-a-time, no other variable-length code will have a shorter expected length.

The algorithm builds the binary tree for the encoding from the bottom up.

Start by choosing the two symbols with the smallest probability (which means they have highest information content and should have the longest encoding).

If anywhere along the way, two symbols have the same probability, simply choose one arbitrarily.

In our running example, the two symbols with the lowest probability are C and D.

Combine the symbols as a binary subtree, with one branch labeled "0" and the other "1".

It doesn't matter which labels go with which branch.

Remove C and D from our list of symbols, and replace them with the newly constructed subtree, whose root has the associated probability 1/6, the sum of the probabilities of its two branches.

Now continue, at each step choosing the two symbols and/or subtrees with the lowest probabilities, combining the choices into a new subtree.

At this point in our example, the symbol A has the probability 1/3, the symbol B the probability 1/2 and the C/D subtree probability 1/6.

So we'll combine A with the C/D subtree.

On the final step we only have two choices left: B and the A/C/D subtree, which we combine in a new subtree, whose root then becomes the root of the tree representing the optimal variable-length code.

Happily, this is the code we've been using all along!

As mentioned above, we can produce a number of different variable-length codes by swapping the "0" and "1" labels on any of the subtree branches.

But all those encodings would have the same expected length, which is determined by the distance of each symbol from the root of the tree, not the labels along the path from root to leaf.

So all these different encodings are equivalent in terms of their efficiency.

Now try your hand at using Huffman's Algorithm to construct optimal variable-length encodings.