Character	Code	Frequency	Total Bits
a	001	10	30
e	01	15	30
i	10	12	24
s	00000	3	15
t	0001	4	16
space	11	13	26
newline	00001	1	5
Total			146

Figure 10.12 Optimal prefix code

Notice that there are many optimal codes. These can be obtained by swapping children in the encoding tree. The main unresolved question, then, is how the coding tree is constructed. The algorithm to do this was given by Huffman in 1952. Thus, this coding system is commonly referred to as a Huffman code.

Huffman's Algorithm

Throughout this section we will assume that the number of characters is C. Huffman's algorithm can be described as follows: We maintain a forest of trees. The *weight* of a tree is equal to the sum of the frequencies of its leaves. $C - 1$ times, select the two trees, T_1 and T_2, of smallest weight, breaking ties arbitrarily, and form a new tree with subtrees T_1 and T_2. At the beginning of the algorithm, there are C single-node trees–one for each character. At the end of the algorithm there is one tree, and this is the optimal Huffman coding tree.

A worked example will make the operation of the algorithm clear. Figure 10.13 shows the initial forest; the weight of each tree is shown in small type at the root. The two trees of lowest weight are merged together, creating the forest shown in Figure 10.14. We will name the new root $T1$, so that future merges can be stated unambiguously. We have made *s* the left child arbitrarily; any tiebreaking procedure can be used. The total weight of the new tree is just the sum of the weights of the

Figure 10.13 Initial stage of Huffman's algorithm

Figure 10.14 Huffman's algorithm after the first merge