The Huffman encoding procedure is simple, where the fixed-length source symbols are mapped onto variable-length codewords as follows.

Step 1: Arrange probabilities of source symbols in the descending order.

Step 2: Merge the two least probable symbols into a pseudo symbol and repeat **Steps 1-2**, until a unity root pseudo symbol is reached.

Step 3: In the result of Huffman tree, label the left arc of each internal node with '1', and right arc with '0' or vice versa.

Step 4: The codeword for each symbol is the sequence of labels along the path from the root to the leaf node representing the symbol.

Figure 2.1 shows an example of the combining process of a Huffman code. the source consists of six symbols from an alphabet $\{s_0, s_1, s_2. s_3. s_4, s_5\} = \{a, b, c, d, e, f\}$

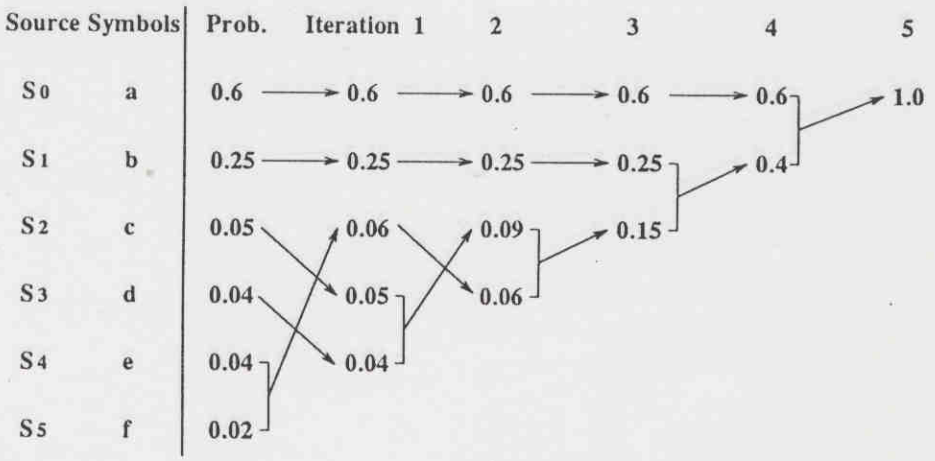


Figure 2.1: The process for building a Huffman code.

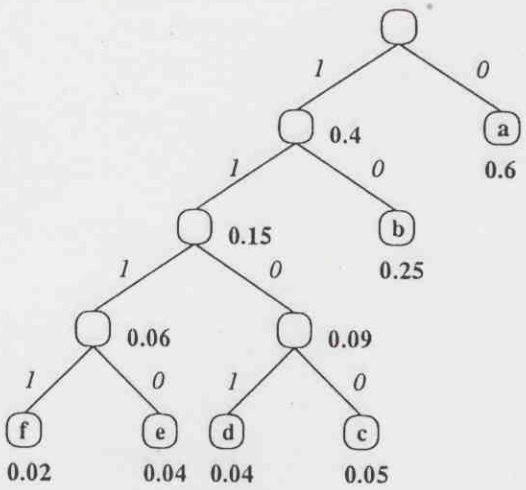
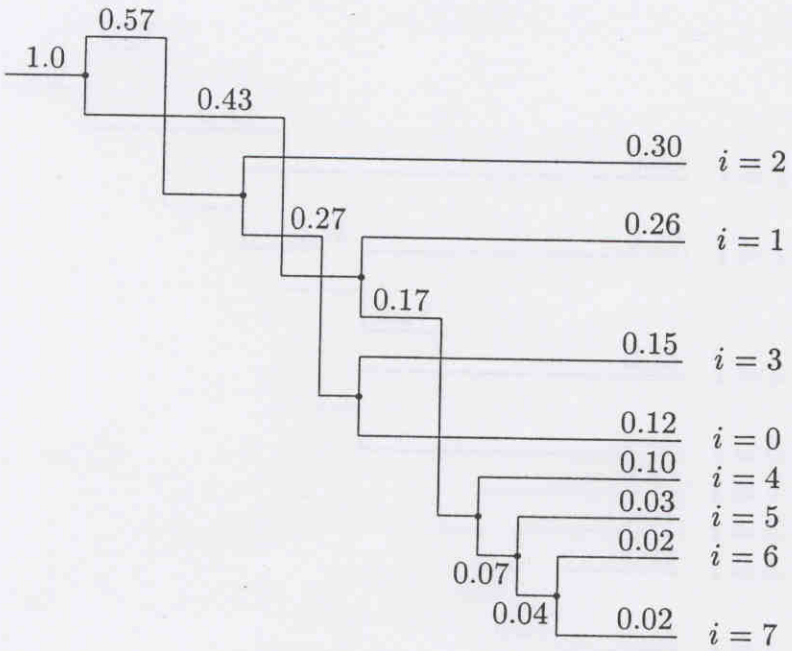


Figure 2.2: A Huffman tree for the example in Figure 2.1.

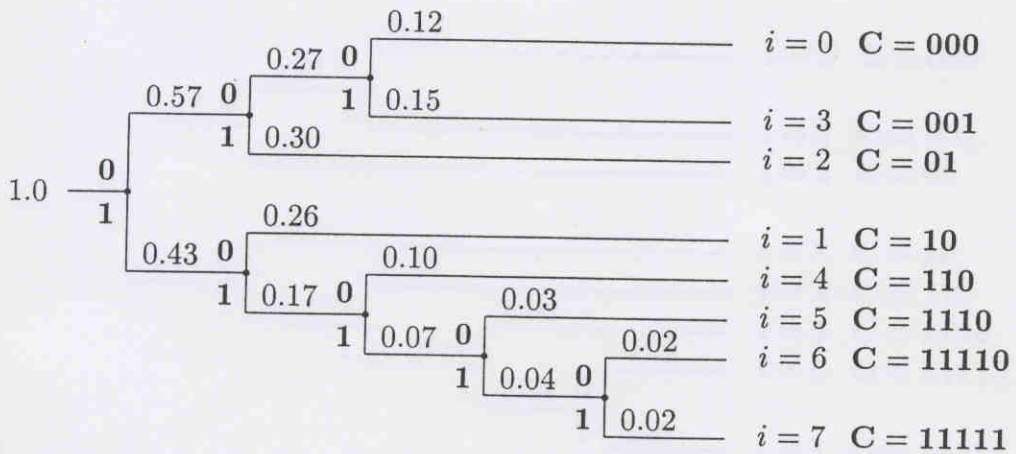
with probabilities 0.6, 0.25, 0.05, 0.04, 0.04, and 0.02, respectively. A result of Huffman tree for the code assignment of source symbols is illustrated in Figure 2.2. The six source symbols can be assigned by the codewords 0. 10, 1100, 1101, 1110, and 1111, respectively. A source symbol with a larger probability is assigned a codeword with a smaller length. The average length of each source symbol after Huffman coding is

$$\begin{aligned}\sum_{i=0}^{5} p(s_i)\, l_i &= 0.6 \times 1 + 0.25 \times 2 + 0.05 \times 4 + 0.04 \times 4 + 0,04 \times 4 + 0.02 \times 4 \\ &= 1.7\ (bits/symbol).\end{aligned}$$

where $p(s_i)$ is the ith symbol probability, l_i means the length of the ith codeword.



(a)



(b)

Fig. 4.2.1 (a) Construction of Huffman code tree; (b) tree rearrangement.