Huffman Encoding

Shiv Shankar Dayal

Theory

Suppose that we have an alphabet of n symbols and a long message consisting of symbols from this alphabet. For example, suppose that the alphabet consists of the four symbols A, B, C, and D) and that codes are assigned to these's ymbols as follows:

Symbol	Code
A	010
B	100
C	000
D	111

The message ABACCDA would then be encoded as 010100010000000111010. Such an encoding is inefficient, since three bits are used for each symbol, so that 21 bits are needed to encode the entire message. Suppose that a two-bit code is assigned to each symbol, as follows:

Symbol	Code
A	00
B	01
C	10
D	11

Then the code for the message would be 00010010101100, which requires only 14 bits.

Each of the letters B and D appears only once in the message, whereas the letter A appears three times. If a code is chosen so that the letter A is assigned a shorter bit string than the letters B and D, the length of the encoded message would be small. This is because the short code (representing the letter A) would appear more frequently than the long code. Indeed, codes can be assigned as follows:

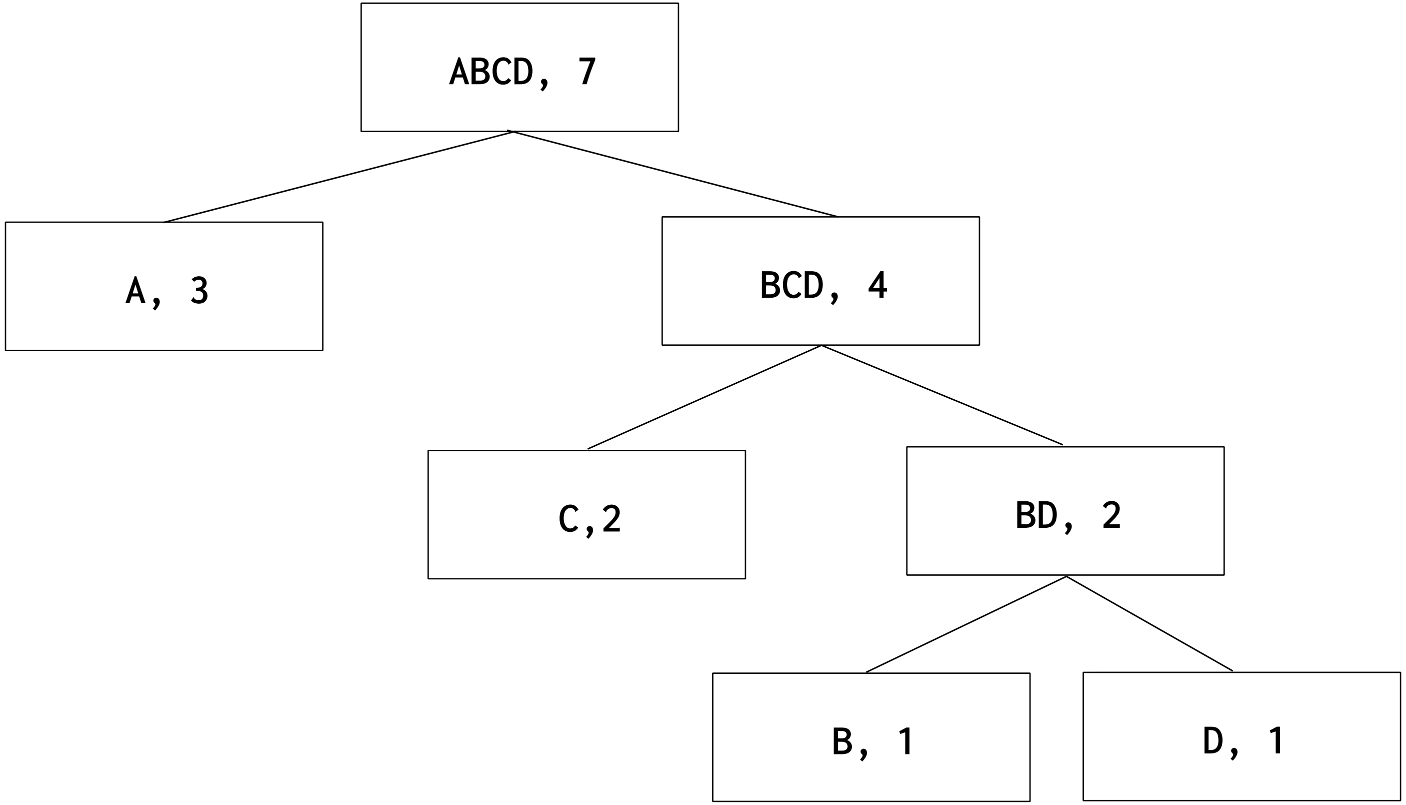
Symbol	Code
A	0
B	110
C	10
D	111

Using this code, the message ABACCDA is encoded as 0110010101110. which requires only 13 hits. In very long messages containin g symbols that appear very infrequentiv. the savings are substantial. Ordinarily, codes are not constructed on the basis of the frer quency of characters within a single message alone. but on the basis of their frequency within a whole set of messages. The same code set is then used for each message.

Find the two symbols that appear least frequently. In our example, these are B and D. The last bit of their codes differentiates one from the other: 0 for B and 1 for D Combine these two symbols into the single symbol BD, whose code represents the knowledge that a symbol is either a B or a D. The frequency of occurrence of this new symbol is the sum of the frequencies of its two constituent symbols. Thus the frequency of BD is 2. There are now three symbols: A (frequency 3), C (frequency 2) and BD (frequency 2). Again choose the two symbols with smallest frequency: C and BD. The last bit of their codes again differentiates one from the other: 0 for C and 1 for BD. The two symbols are then combined into the single symbol CBD with frequency 4. There are now only two symbols remaining: A and CBD. These are combined into the single symbol ACBD. The last bits of the codes for A and CBD differentiate one front other: 0 for A and 1 for CBD.

The symbol ACBD contains the entire alphabet; it is assigned the null hit string of length 0 as its code. At the start of the decoding, before any bits have been examined, it is ce1ain that any symbol is contained in ACBD. The two symbols that make up ACBD (A and CBD) are assigned the codes 0 and 1, respectively. If a 0 is encountered, the encoded symbol is an A; if a 1 is encountered, it is a C, a B, or a D. Similarly, the two symbols that constitute CBD (C and SD) are assigned the codes 10 and 11. respectively. The first bit indicates that the symbol is one of the constituents of CBD, and the second bit indicates whether it is a C or a BD. The symbols that make up BD (B and D) are then assigned the codes 110 and 111.

The action of combining two symbols into one suggests the use of a binary tree. Each node of the tree represents a symbol and each leaf represents a symbol of the orig- inal alphabet. Such trees are callled *Huffman* trees.



ABCD, 7

A, 3

BCD, 4

C,2

BD, 2

B, 1

D, 1