This lecture explains how to manipulate bits. The topic is bit wise operations.

The purpose of Huffman compression is to reduce the sizes of files. However, the method we have explained so far will actually increase file sizes.

The format of a compressed file includes three parts: the code tree, the length of the original file, and the code of the data.

Let’s consider the third part only. If the input is ei, ei, cee, ass, #, M, gee, cee, ass, ei, etc. .

The code for the data will be zero,zero, zero,zero, one,zero, one,one, zero,one,one,zero, zero,one,zero, zero,one,one,one, one,zero, one,one, zero,zero etc. .

The input data needs only 10 bytes. The code needs 25 bytes. This does not compress the file.

What we need to do is to use only one bit for each one or zero. One bit can express either one or zero.

One byte has 8 bits and can express a number between zero and 255.

If each zero or one is expressed by only one bit, then the first four characters can be expressed by only one byte. Thus, we can reduce the length of four bytes into only one byte.

This is compression.

Unfortunately, most programming languages, including C, do not support the bit data type. Instead, the minimum unit is a byte. C programs use unsigned char to store a byte. We need to learn bit wise operations to manipulate bits. This lecture explains four bit wise operations: bit wise and, bit wise or, shift right, and shift left.

What is bit wise and? Consider two input bits. Each can be either zero or one. The output is one only if both input bits are one. If one or both inputs are zero, the output is zero.

Next, let’s consider bit wise or. Each input can be either zero or one. The output is zero if both input bits are zero. If one or both inputs are one, the output is one.

Before we talk about how to use the bit wise operations, let’s review the number systems.

Hexadecimal system uses 16 as the base and has 16 digits. The 16 digits include zero to 9. Additional 6 digits are needed and they are ei, bee, cee, dee, E, and F. .

We use subscripts with parentheses to indicate the bases. The first example is one two three four using ten as the base. What does this mean?

We should start from the right end. Four means four times ten with power of zero. Three four means three times ten with power of one, plus four times ten with power of zero.

The binary number one zero one one means one times two with power of three, plus zero times two with power of two, plus one times two with power of one, plus one time two with power of zero.

.

The hexadecimal number B 9 C 6 means eleven times sixteen with power of three, plus nine times sixteen with power of two, plus twelve times sixteen with power of one, plus six time sixteen with power of zero.

We can also express numbers that are smaller than the base.

In the first example, a decimal number 5 one 2 point 3 4 means 5 times 10 with power of 2, plus one times 10 with power of one, plus 2 times 10 with power of zero. Point 3 4 means 3 times 10 with power of minus one, plus 4 times 10 with power of minus 2.

Let’s consider two bytes. Ei is hexadecimal 3 Dee. Bee is hexadecimal nine six. If a number starts with zero X, it is written in the hexadecimal format.

Please notice that you should not use zero bee to express the binary format. This is not supported in standard cee language.

If we write ei in the binary format, it is zero, zero, one, one, one, one, zero, one. If we write bee in the binary format, it is one,zero,zero,one, zero,one,one,zero.

If cee is ei and bee, what is the value of cee?

The left most bit is called the most significant bit, or M. S. B.. The rightmost bit is called the least significant bit, or L. S. B. .

Let’s start from the leftmost bit. For ei, this bit is zero. Thus, this bit is zero for cee.

For bee, this bit is one. Thus, this bit is one for dee.

Please remember that if the operation is and, the output is zero if one of the inputs is zero. If the operation is or, the outputs is one if one of the input is one.

Let’s consider the next bit. Both ei and bee have zero in this bit. Thus, this bit is zero for both cee and dee.

Let’s move to consider the third bit from the left. Ei has one in this bit and bee has zero in this bit. Cee has zero in this bit and dee has one in this bit.

We move to the fourth bit from the left. Both ei and bee have one in this bit. Thus, this bit is one for both cee and dee.

We can continue in the other four bits. The value of cee is binary zero,zero,zero,one, zero,one,zero,zero. If it is expressed in hexadecimal, it is one four. The value of dee is binary one,zero,one,one,one,one,one,one,. If it is expressed hexadecimal, it is bee F. .

Next, let’s study how shift works. We can shift left or shift right.

Consider cee is ei shift left by two. We take the bits from ei. The leftmost two bits are discarded.

Every bit moves left by two bits.

Two zeros are added to the right end of cee.

The value of cee expressed in binary is one,one,one,one,zero,one,zero,zero. Expressed in hexadecimal, it is F, four.

The value of dee is bee shift right by three bits.

The three bits at the right end of bee are discarded.

Every bit moves right by three bits.

Three zero bits are added to the left end of bee.

The binary value of dee is zero,zero,zero,one,zero, zero,one,zero. In the hexadecimal format, it is one two.