In [information theory](http://en.wikipedia.org/wiki/Information_theory), the **Hamming distance** between two [strings](http://en.wikipedia.org/wiki/String_%28computer_science%29) of equal length is the number of positions at which the corresponding symbols are different. Put another way, it measures the minimum number of *substitutions* required to change one string into the other, or the number of *errors* that transformed one string into the other.

Hamming distance can calculate just rely on the hamming distance. Nothing about the bit of the length of the input x,y.

The following [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) function will compute the Hamming distance of two integers (considered as binary values, that is, as sequences of bits). The running time of this procedure is proportional to the Hamming distance rather than to the number of bits in the inputs. It computes the [bitwise](http://en.wikipedia.org/wiki/Bitwise_operation) [exclusive or](http://en.wikipedia.org/wiki/Exclusive_or) of the two inputs, and then finds the [Hamming weight](http://en.wikipedia.org/wiki/Hamming_weight) of the result (the number of nonzero bits) using an algorithm of [Wegner (1960](http://en.wikipedia.org/wiki/Hamming_distance#CITEREFWegner1960)) that repeatedly finds and clears the lowest-order nonzero bit.

思想：

Val 为二进制补码类型充分必要条件， val 的位全为0，则val==0；反之也成立；

当val != 0 时候，说明val里面至少有一个1，把这个1去掉(val &= val -1)，dist自加，然后再检查val；

unsigned hamdist(unsigned x, unsigned y)

{

unsigned dist = 0, val = x ^ y;

// Count the number of set bits

while(val)

{

++dist;

val &= val - 1;

}

return dist;

}