Jdk8中HashMap获取key对应的Node是以如下的方式：

tab[(n - 1) & (hash = *hash*(key))])

其中n是内部数组长度.

实际上是一种高效的数组取余法

即(n-1) & hash(key) == hash(key) % n

这是为什么呢？

首先有一个限制条件， n必须是2的正整数指数（当初始化指定的initialCapacity不是2的正整数指数形式时，也会将转换为2的指数形式），代码如下：

*/\*\*  
 \* Returns a power of two size for the given target capacity.  
 \*/*static final int tableSizeFor(int cap) {  
 int n = -1 >>> Integer.*numberOfLeadingZeros*(cap - 1);  
 return (n < 0) ? 1 : (n >= *MAXIMUM\_CAPACITY*) ? *MAXIMUM\_CAPACITY* : n + 1;  
}

由于n是2的指数， 因此n的2进制为10000…(n个0). n-1则为低n-1位全部为1， 大于等于n的位置全部为0。 而用来做取余的数是对n做取余， 在大于等于n的位置显然是可以除尽的，只需要就考虑小于n的位置，即0到n-1位， 因此与n-1做&运算恰好可以得出做取余的数的低n-1位，即余数。