

# 位运算技巧

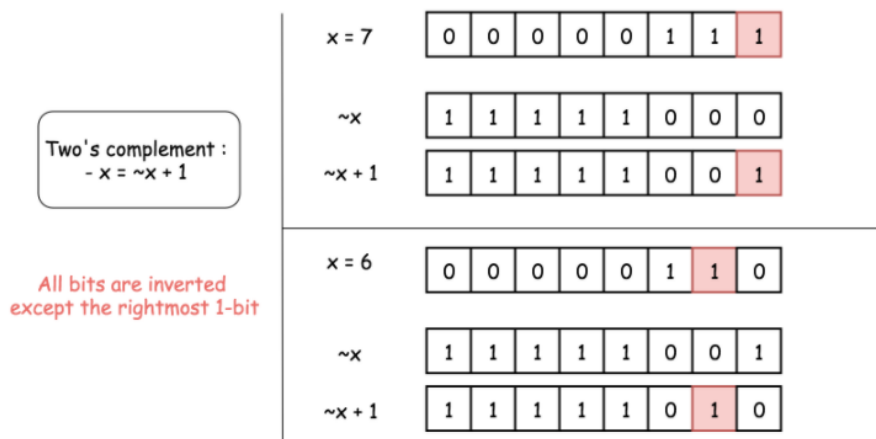
## $x \& (-x)$ 用于取最右边的1;

首先讨论为什么  $x \& (-x)$  可以获取到二进制中最右边的 1，且其它位设置为 0。

在补码表示法中， $-x = \neg x + 1$ 。换句话说，要计算  $-x$ ，则要将  $x$  所有位取反再加 1。

在二进制表示中， $\neg x + 1$  表示将该 1 移动到  $\neg x$  中最右边的 0 的位置上，并将所有较低位的位设置为 0。而  $\neg x$  最右边的 0 的位置对应于  $x$  最右边的 1 的位置。

总而言之， $-x = \neg x + 1$ ，此操作将  $x$  所有位取反，但是最右边的 1 除外。



## $x \& (x-1)$ 用于消除最右边的1

## 利用与一个特定的数进行与运算达到确定一个数多个位的情况

```
x & 0x0000000f; // 用于提取x的前四位;
```