

二进制位运算

2014 年 2 月 22 日

位运算的简单运用(摘自matrix博客)

有时我们的程序需要一个规模不大的Hash表来记录状态。比如，做数独时我们需要27个Hash表来统计每一行、每一列和每一个小九宫格里已经有哪些数了。此时，我们可以用27个小于 2^9 的整数进行记录。例如，一个只填了2和5的小九宫格就用数字18表示（二进制为000010010），而某一行的状态为511则表示这一行已经填满。需要改变状态时我们不需要把这个数转成二进制修改后再转回去，而是直接进行位操作。在搜索时，把状态表示成整数可以更好地进行判重等操作。这道题是在搜索中使用位运算加速的经典例子。以后我们会看到更多的例子。下面列举了一些常见的二进制位的变换操作。

功能	示例	位运算
去掉最后一位	(101101->10110)	x shr 1
在最后加一个0	(101101->1011010)	x shl 1
在最后加一个1	(101101->1011011)	x shl 1+1
把最后一位变成1	(101100->101101)	x or 1
把最后一位变成0	(101101->101100)	x or 1-1
最后一位取反	(101101->101100)	x xor 1
把右数第k位变成1	(101001->101101,k=3)	x or (1 shl (k-1))
把右数第k位变成0	(101101->101001,k=3)	x and not (1 shl (k-1))
右数第k位取反	(101001->101101,k=3)	x xor (1 shl (k-1))
取末三位	(1101101->101)	x and 7
取末k位	(1101101->1101,k=5)	x and (1 shl k-1)
取右数第k位	(1101101->>1,k=4)	x shr (k-1) and 1
把末k位变成1	(101001->101111,k=4)	x or (1 shl k-1)
末k位取反	(101001->100110,k=4)	x xor (1 shl k-1)

把右边连续的1变成0	(100101111->100100000)	x and (x+1)
把右起第一个0变成1	(100101111->100111111)	x or (x+1)
把右边连续的0变成1	(11011000->11011111)	x or (x-1)
取右边连续的1	(100101111->1111)	(x xor (x+1)) shr 1
去掉右起第一个1的左边	(100101000->1000)	x and (x xor (x-1))