**201. Bitwise AND of Numbers Range**

置顶

Created At : 2021-04-08 00:03

Count:794 Views 👀 :10 [Comment:0](https://grandyang.com/leetcode/201/#comments)

Given a range [m, n] where 0 <= m <= n <= 2147483647, return the bitwise AND of all numbers in this range, inclusive.

For example, given the range [5, 7], you should return 4.

Credits:  
Special thanks to [@amrsaqr](https://leetcode.com/discuss/user/amrsaqr) for adding this problem and creating all test cases.

又是一道考察位操作Bit Operation的题，相似的题目在LeetCode中还真不少，比如[Repeated DNA Sequences 求重复的DNA序列](http://www.cnblogs.com/grandyang/p/4284205.html)，[Single Number 单独的数字](http://www.cnblogs.com/grandyang/p/4130577.html),  [Single Number II 单独的数字之二](http://www.cnblogs.com/grandyang/p/4263927.html) ，[Grey Code 格雷码](http://www.cnblogs.com/grandyang/p/4315649.html)，和[Reverse Bits 翻转位](http://www.cnblogs.com/grandyang/p/4321355.html) 等等，那么这道题其实并不难，我们先从题目中给的例子来分析，[5, 7]里共有三个数字，分别写出它们的二进制为：

101　　110　　111

相与后的结果为100，仔细观察我们可以得出，最后的数是该数字范围内所有的数的左边共同的部分，如果上面那个例子不太明显，我们再来看一个范围[26, 30]，它们的二进制如下：

11010　　11011　　11100　　11101　　11110

发现了规律后，我们只要写代码找到左边公共的部分即可，我们可以从建立一个32位都是1的mask，然后每次向左移一位，比较m和n是否相同，不同再继续左移一位，直至相同，然后把m和mask相与就是最终结果，代码如下：

解法一：

class Solution {

public:

int rangeBitwiseAnd(int m, int n) {

int d = INT\_MAX;

while ((m & d) != (n & d)) {

d <<= 1;

}

return m & d;

}

};

复制代码

此题还有另一种解法，不需要用mask，直接平移m和n，每次向右移一位，直到m和n相等，记录下所有平移的次数i，然后再把m左移i位即为最终结果，代码如下：

解法二：

class Solution {

public:

int rangeBitwiseAnd(int m, int n) {

int i = 0;

while (m != n) {

m >>= 1;

n >>= 1;

++i;

}

return (m << i);

}

};

复制代码

下面这种方法有点叼，就一行搞定了，通过递归来做的，如果n大于m，那么就对m和n分别除以2，并且调用递归函数，对结果再乘以2，一定要乘回来，不然就不对了，就举一个最简单的例子，m = 10, n = 11，注意这里是二进制表示的，然后各自除以2，都变成了1，调用递归返回1，这时候要乘以2，才能变回10，参见代码如下：

解法三：

class Solution {

public:

int rangeBitwiseAnd(int m, int n) {

return (n > m) ? (rangeBitwiseAnd(m / 2, n / 2) << 1) : m;

}

};

复制代码

下面这种方法也不错，也很简单，如果m小于n就进行循环，n与上n-1，那么为什么要这样与呢，举个简单的例子呗，110与上(110-1)，得到100，这不就相当于去掉最低位的1么，n就这样每次去掉最低位的1，如果小于等于m了，返回此时的n即可，参见代码如下：

解法四：

class Solution {

public:

int rangeBitwiseAnd(int m, int n) {

while (m < n) n &= (n - 1);

return n;

}

};

复制代码

参考资料：

<https://discuss.leetcode.com/topic/13508/one-line-c-solution>

<https://discuss.leetcode.com/topic/12133/bit-operation-solution-java>

<https://discuss.leetcode.com/topic/20176/2-line-solution-with-detailed-explanation>

[LeetCode All in One 题目讲解汇总(持续更新中…)](http://www.cnblogs.com/grandyang/p/4606334.html)

*转载请注明来源于 Grandyang 的博客 (grandyang.com)，欢迎对文章中的引用来源进行考证，欢迎指出任何有错误或不够清晰的表达。可以在下面评论区评论，也可以邮件至 grandyang@qq.com*