这道题目的思路如果是逐个数去数显然是会超时的。

解决这个问题的关键在于一个结论：

N位，使用l个1组成的数的个数为c[n][l]，则其可以分为两部分，一部分是第n位为1，一部分是第n位为0：

C[n][l]=c[n-1][l-1]+c[n-1][l]。这两部分数不会重合，则可以通过DP计算出这个32\*32的数组。

对于第I个数，首先看c[i][L]的集合，这个集合里面的最大的数就是(i-L…i)位都是1，余下的位都是0的这个数a。那么a就是第c[i][L]大的数，因为其他的数最高位必须超过i，则肯定大于这个集合里面的所有数。

因此，I小于c[n][l]意味着他在这个集合里面，而这个集合又能够被分为第n位是1的集合和第n位不是1的集合。而I大于c[n-1][l]则意味着他在第n位为1的集合里面。

这样，这个数字的第n位就一定是1.至于他在第n位为1的集合里面的第几个呢？

由于第n位为1的集合严格大于第n位为0的集合，因此，他在这个集合的位置就是x-（第n位为0的集合），而这样确认了第i位后，在i-1位到以前的所有数中，变成了一个子问题：从n-1,l-1组成的数里面找出第x-c[i-1][l-1]大的数，以此递归，直到x足够小（1或者2），而我们知道，x=1时，这个数是0，x=2时，这个数是1.