~~不知为什么这题我一看到就一下子会做了。。。~~

我的做法：

考虑按位确定答案，于是我们要能算出对于一个确定的前缀，有多少个合法的数是以这个前缀开头的。

首先我们可以观察到一个性质：所有的答案的第一位肯定是0，否则它的取反串肯定比他小。在确定了第一位是0之后，我们之后在计算个数的时候就不用考虑取反串和它本身的大小问题了

第二我们可以发现当前串的最后一位如果是0，那么原串的取反串的倒串就肯定比原串小；最后一位如果是1，那么原串的倒串肯定比原串小，所以一旦确定了最后一位，我们就又可以少考虑一种情况。

所以在计算个数的时候，我们先枚举最后一位是0还是1，这样我们自始至终就只要考虑原串和其倒串的大小关系或者是原串和其取反倒串的关系。

先讨论最后一位是0的情况，我们考虑按位枚举后缀，后缀的每一位和前缀的一位相对应。如果前缀的对应位是0，那么后缀的对应位放1就肯定是合法的了，中间可以2的若干次方直接计算答案，当然也可以放0继续和前缀匹配，如果前缀的对应位是1，就只能放1和其匹配。

设当前前缀长度是*len*,我们匹配的总位数是*Len*=*min*(*len*,*n*−*len*),全部匹配完之后，我们又要分两种情况讨论：

1. Len>n/2此时整个串已经完全确定，我们还需要检验一下中间的这些位是否满足条件，这个部分比较简单。
2. Len<n/2,此时我们中间还有一些空位要填，考虑如何快速计算有多少种填法符合条件，我们可以发现，如果中间填一个回文串肯定是符合条件的，回文串的个数很好计算，相当于将对应的位打包，个数是2^((n-len-len+1)/2),对于不是回文串的串，我们发现一个合法的填法取反肯定是不合法的，一个不合法的填法取反肯定是合法的，所以合法的和不合法的填法一一对应，我们用总填法数减去回文的填法数除以二即可

再讨论最后一位是1的情况，大部分和前面是一样的。我们可以一样的按位和前缀匹配，匹配完了之后*len*≥*n/2*的做法也是一样的，但在*len*<*n/2*的时候，非回文的填法还是那样算，但我们发现回文的填法并不全是合法的，如果*n*−*Len*−*Len*是偶数是没有问题的，如果是奇数，我们会发现，最中间的那个单独的一位只能是0不能是1，所以这是回文的合法个数等于回文的总个数除以二。

时间复杂度*O*(*n^*2)

这题做法很多，比如说看到n<=50，显然会往指数级的算法上想，比如说tourist的做法,考虑从两端向中间确定答案，四种情况中有一种肯定不合法，有一种可以直接算，另两种直接搜，复杂度就是2^(n/2)\*n，可以通过，很好写。

再比如楼教主的做法是在搜的过程中压一个三位Mask，表示当前串的reverse串，取反串和取反reverse串是否会“构成威胁”，一旦是0就直接2的次幂算,写起来也很方便，不容易错。

有的做法复杂度可能不好，但很好写，在n的范围较小的时候不妨尝试这些方法。比如我自己想的方法中分类讨论有好几个细节，很容易写错，比如如果不做长度为奇数的样例最后一个情况就会漏掉，虽然快但对思维要求较高。