程序员的算法趣题之Q01：回文十进制数

**什么是回文数？**

如果把某个数的各个数字按相反的顺序排列，得到的数和原来的数相同，则这个数就是“回文数”。譬如123454321就是一个回文数。

**那么问题来了……**

求用十进制、二进制、八进制表示都是回文数的所有数字中，大于十进制数10的最小值。

例如：9（十进制数）=1001（二进制数）=11（八进制数）

\*\*本例中的十进制数9小于10，因此不符合要求。

**关于二进制**

与十进制使用0,1,2,3,4,5,6,7,8,9这10个数字表示数的方式类似，二进制只使用0和1来表示数，也就是说，二进制的所有数只包含0和1这两个数字，比如0,1,10,100,101,1000，…

十进制数3984由3个1000（=10^3）、9个100（=10^2）、8个100（=10^1）和4个1（=10^0）组成。二进制数1011由1个8（=2^3）、0个4（=2^2）、1个2（=2^1）和1个1（=2^0）组成。

也就是说，二进制数1011是8+0+2+1，对应十进制数11。反过来，知道十进制数要求二进制数时，则是用这个数除以2，得到的商再除以2，再用这一步得到的商除以2，直到商变为0,。最后，把过程中得到的余数逆序排列就能得到相应的二进制数。

举个例子，如果给定十进制数19，则求对应的二进制数的过程如下。

19÷2=9余1

9÷2=4余1

4÷2=2余0

2÷2=1余0

1÷2=0余1

从下往上排列余数后就可以得到二进制数10011。

==================分割线===================

思路

因为是二进制的回文数，所以如果最低位是0，那么相应地最高位也是0.但是，以0开头肯定是不恰当的，由此可知最低位为1。（有时候我们会看到以0开头的二进制数，其实前面的0可以省略）

如果用二进制表示时最低位为1，那这个数一定是奇数（因为二进制的位数从低到高分别代表着1,2,4,8,16,32,64，…，所以最低位是1或0直接决定二进制数是奇数还是偶数），因此只考虑奇数的情况就可以。接下来可以简单地编写程序，从10的下一个数字11开始，按顺序搜索。

Python代码如下

**def** isHuiwen(number):  
 dec\_num = str(number)  
 *# python将十进制数转为二进制或八进制数时会带有前缀0b, 0o，所以要将前缀去掉* bin\_num = str(bin(number))[2:]  
 oct\_num = str(oct(number))[2:]  
 *# 要获得字符串s的逆序可用s[::-1]  
 # 如果三个条件都满足，则返回True,否则返回False* **return** dec\_num == dec\_num[::-1] **and** bin\_num == bin\_num[::-1] **and** oct\_num == oct\_num[::-1]  
  
  
num = 11  
**while True**:  
 *# 调用函数判断num是否是回文数，如果是，则输出该数，并停止循环* **if** isHuiwen(num):  
 print(num)  
 **break** num += 2

参考《程序员的算法趣题》