```
1 /*
  我们称一个数 X 为好数, 如果它的每位数字逐个地被旋转 180 度后,我们仍可以得到一个有效的,
   且和 X 不同的数。要求每位数字都要被旋转。
   如果一个数的每位数字被旋转以后仍然还是一个数字, 则这个数是有效的。0, 1, 和 8 被旋转后仍
   然是它们自己; 2 和 5 可以互相旋转成对方; 6 和 9 同理,除了这些以外其他的数字旋转以后都不
   再是有效的数字。
  现在我们有一个正整数 N, 计算从 1 到 N 中有多少个数 X 是好数?
4
5
   示例:
   输入: 10
7
  输出: 4
  解释:
8
9
  在[1, 10]中有四个好数: 2, 5, 6, 9。
10 注意 1 和 10 不是好数, 因为他们在旋转之后不变。
11
   N 的取值范围是 [1, 10000]。
12
13 来源: 力扣 (LeetCode)
  链接: https://leetcode-cn.com/problems/rotated-digits
14
15 著作权归领扣网络所有。商业转载请联系官方授权,非商业转载请注明出处。
16 */
```

分析:

• 依次获取n的每一位,其每一位都必须在0,1,8,2,5,6,9,且为保证翻转后的数与原来数据不一样,至少有一位在2,5,6,9中.

方法一:C++

```
1 class Solution
 2
   {
 3
 4
        private:
 5
            bool isGoodNum(int n)
            {
 6
                //int num_1 = 0; /*0 1 8 2 5 6 9 个数*/
 7
 8
                int num_2 = 0; /*2 5 6 9 个数*/
                while(n)
9
10
                 {
11
                     switch(n%10)
12
                     {
13
                         case 2:
14
                         case 5:
15
                         case 6:
16
                         case 9:
17
                             //num_1++;
18
                             num_2++;
19
                         break;
20
21
                         case 0:
22
                         case 1:
23
                         case 8:
```

```
24
                       //num_1++;
                     break;
25
26
27
                     default:
28
                        return false;
29
                  }
30
                  n /= 10;
31
              }
32
33
             return (num_2>0);
34
          }
35
36
      public:
37
38
          int rotatedDigits(int N)
39
40
              int ret_val = 0;
41
42
              for(int i=1;i <= N;i++ )</pre>
43
44
                 if(isGoodNum(i))
45
46
                    ret_val++;
47
                  }
48
49
              return ret_val;
50
          }
51 };
52 /*
53 执行结果:
54 通过
55 显示详情
56 执行用时 :8 ms, 在所有 C++ 提交中击败了63.34% 的用户
57 内存消耗 :8.2 MB, 在所有 C++ 提交中击败了82.81%的用户
58 */
```

AlimyBreak 2019.09.14