⭐🖌 788 Rotated Digits

2018年5月4日　　20:45

X is a good number if after rotating each digit individually by 180 degrees, we get a valid number that is different from X. Each digit must be rotated - we cannot choose to leave it alone.

A number is valid if each digit remains a digit after rotation. 0, 1, and 8 rotate to themselves; 2 and 5 rotate to each other; 6 and 9 rotate to each other, and the rest of the numbers do not rotate to any other number and become invalid.

Now given a positive number N, how many numbers X from 1 to N are good?

Example:

Input: 10

Output: 4

Explanation:

There are four good numbers in the range [1, 10] : 2, 5, 6, 9.

Note that 1 and 10 are not good numbers, since they remain unchanged after rotating.

Note:

- N will be in range [1, 10000].

来自 <https://leetcode.com/problems/rotated-digits/description/>

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?

示例:

输入: 10

输出: 4

解释:

在[1, 10]中有四个好数： 2, 5, 6, 9。

注意 1 和 10 不是好数, 因为他们在旋转之后不变。

注意:

- N 的取值范围是 [1, 10000]。

Solution for Python3:

```python
class Solution1:
    def rotatedDigits(self, N):
        """
        :type N: int
        :rtype: int
        """
        def goodnumber(n):
            isgood = False
            while n:
                m = n % 10
                if m in [2,5,6,9]:
                    isgood = True
                elif m in [3,4,7]:
                    return False
                n //= 10
            return isgood
        return sum(goodnumber(n) for n in range(1,N+1))

class Solution2:
    def rotatedDigits(self, N):
        """
        :type N: int
        :rtype: int
        """
        dp, cnt = [0]*(N+1), 0
        for i in range(N+1):
            if i < 10:
                if i == 0 or i == 1 or i == 8:
                    dp[i] = 1
                elif i == 2 or i == 5 or i == 6 or i == 9:
                    dp[i] = 2
                    cnt += 1
            else:
                a, b = dp[i // 10], dp[i % 10]
                if a == b == 1:
                    dp[i] = 1
                elif a >= 1 and b >= 1:
                    dp[i] = 2
                    cnt += 1
        return cnt
```

Solution for C++:

```cpp
class Solution1 {
public:
    int rotatedDigits(int N) {
        int ans = 0;
        for (int i = 1; i <= N; i++)
            if (goodnumber(i))
                ans++;
        return ans;
    }
    bool goodnumber(int n) {
        bool isgood = false;
        while (n) {
            int m = n % 10;
            if (m == 2 || m == 5 || m == 6 || m == 9)
                isgood = true;
            else if (m == 3 || m == 4 || m == 7)
                return false;
            n /= 10;
        }
        return isgood;
    }
};

// dp[i] = 0, invalid number
// dp[i] = 1, valid and same number
// dp[i] = 2, valid and different number
class Solution2 {
public:
    int rotatedDigits(int N) {
        int dp[N+1] = {0};
        int cnt = 0;
        for (int i = 0; i <= N; i++) {
            if (i < 10) {
                if (i == 0 || i == 1 || i == 8)
                    dp[i] = 1;
                else if (i == 2 || i == 5 || i == 6 || i == 9) {
                    dp[i] = 2;
                    cnt++;
                }
            } else {
                int a = dp[i / 10], b = dp[i % 10];
                if (a == 1 && b == 1)
                    dp[i] = 1;
                else if (a >= 1 && b >= 1) {
                    dp[i] = 2;
                    cnt++;
                }
            }
        }
        return cnt;
    }
};
```