258 Add Digits

2018年4月7日 18:02

Question:

Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.
For example:
Given num = 38, the process is like: 3 + 8 = 11, 1 + 1 = 2. Since 2 has only one digit, return it.
Follow up:
Could you do it without any loop/recursion in O(1) runtime?

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

给一个非负整数 num，反复添加所有的数字，直到结果只有一个数字。
例如:
设定 num = 38，过程就像： 3 + 8 = 11, 1 + 1 = 2。 由于 2 只有1个数字，所以返回它。
进阶:
你可以不用任何的循环或者递归算法，在 O(1) 的时间内解决这个问题么？

Solution for Python3:

```python
class Solution1:
    def addDigits(self, num):
        """
        :type num: int
        :rtype: int
        """
        while num > 9:
            num = reduce(lambda x, y: x + int(y), str(num), 0)
        return num


class Solution2:
    def addDigits(self, num):
        """
        :type num: int
        :rtype: int
        """
        return num and 1 + (num - 1) % 9
```

Solution for C++:

```cpp
class Solution {
public:
    int addDigits(int num) {
        return 1 + (num - 1) % 9;
```

```
5         }
6     };
```

Appendix:

digital root problem

The problem, widely known as *digit root* problem, has a congruence formula:
https://en.wikipedia.org/wiki/Digital_root#Congruence_formula
For base *b* (decimal case *b* = 10), the digit root of an integer is:

- dr(*n*) = 0 if *n* == 0
- dr(*n*) = (*b*-1) if *n* != 0 and *n* % (*b*-1) == 0
- dr(*n*) = *n* mod (*b*-1) if *n* % (*b*-1) != 0

or

- dr(*n*) = 1 + (*n* - 1) % 9

Note here, when *n* = 0, since (*n* - 1) % 9 = -1, the return value is zero (correct).

来自 <https://leetcode.com/problems/add-digits/discuss/68580/Accepted-C++-O(1)-time-O(1)-space-1-Line-Solution-with-Detail-Explanations>