202 Happy Number

2018年4月4日 15:40

Question:

Write an algorithm to determine if a number is "happy".
A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers.
Example: 19 is a happy number

- $1^2 + 9^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1$

来自 <https://leetcode.com/problems/happy-number/description/>
写一个算法来判断一个数是不是“快乐数”。
一个数是不是快乐是这么定义的：对于一个正整数，每一次将该数替换为它每个位置上的数字的平方和，然后重复这个过程直到这个数变为 1，或是无限循环但始终变不到 1。如果可以变为 1，那么这个数就是快乐数。
案例：19 是一个快乐数。

- $1^2 + 9^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1$

Solution for Python3:

```python
from functools import reduce
class Solution:
    def isHappy(self, n):
        """
        :type n: int
        :rtype: bool
        """
        history = []
        while n not in history:
           history.append(n)
           n = reduce(lambda x,y: x + y**2, map(int,str(n)),0)
           if n == 1:
               return True
        return False

class Solution2:
    def isHappy(self, n):
        """
        :type n: int
        :rtype: bool
        """
        slow, fast = n, n
        while True:
           slow = self.getSquareSum(slow)
```

```
            fast = self.getSquareSum(fast)
            fast = self.getSquareSum(fast)
            if slow == fast:
                break;
        if slow == 1:
            return True
        return False

    def getSquareSum(self, n):
        return sum([int(i)**2 for i in str(n)]);
```

Solution for C++:

```
class Solution1 {
public:
    bool isHappy(int n) {
        unordered_set<int> set;
        int squareSum, remainder;
        while (set.insert(n).second) {
            while (n) {
                remainder = n % 10;
                squareSum += remainder * remainder;
                n /= 10;
            }
            if (squareSum == 1) {
                return true;
            } else {
                n = squareSum;
            }
        }
        return false;
    }
};

class Solution2 {
public:
    bool isHappy(int n) {
        int slow = n, fast = n;
        do {
            slow = getSquareSum(slow);
            fast = getSquareSum(fast);
            fast = getSquareSum(fast);
        } while (slow != fast);
        if (slow == 1) {
            return true;
        } else {
            return false;
        }
    }
    int getSquareSum(int n) {
        int sum = 0, tmp = 0;
        while (n) {
```

```
            tmp = n % 10;
            sum += tmp * tmp;
            n /= 10;
        }
        return sum;
    }
};
```