Happy Number

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

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

Credits:
Special thanks to @mithmatt and @ts for adding this problem and creating all test cases.

Solution 1

I see the majority of those posts use hashset to record values. Actually, we can simply adapt the Floyd Cycle detection algorithm. I believe that many people have seen this in the Linked List Cycle detection problem. The following is my code:

```
int digitSquareSum(int n) {
    int sum = 0, tmp;
    while (n) {
        tmp = n % 10;
        sum += tmp * tmp;
        n /= 10;
    }
    return sum;
}

bool isHappy(int n) {
    int slow, fast;
    slow = fast = n;
    do {
        slow = digitSquareSum(slow);
        fast = digitSquareSum(fast);
        fast = digitSquareSum(fast);
    } while(slow != fast);
    if (slow == 1) return 1;
    else return 0;
}
```

written by Freezen original link here

Solution 2

```
public class Solution {
    public boolean isHappy(int n) {
        int x = n;
        int y = n;
        while(x>1){
            x = cal(x) ;
            if(x==1) return true ;
            y = cal(cal(y));
            if(y==1) return true ;

            if(x==y) return false;
        }
        return true ;
    }
    public int cal(int n){
        int x = n;
        int s = 0;
        while(x>0){
            s = s+(x%10)*(x%10);
            x = x/10;
        }
        return s ;
    }
}
```

written by ibmtp380 original link here

Solution 3

The idea is to use one hash set to record sum of every digit square of every number occurred. Once the current sum cannot be added to set, return false; once the current sum equals 1, return true;

```
public boolean isHappy(int n) {
    Set<Integer> inLoop = new HashSet<Integer>();
    int squareSum,remain;
    while (inLoop.add(n)) {
        squareSum = 0;
        while (n > 0) {
            remain = n%10;
            squareSum += remain*remain;
            n /= 10;
        }
        if (squareSum == 1)
            return true;
        else
            n = squareSum;

    }
    return false;

}
```

written by mo10 original link here