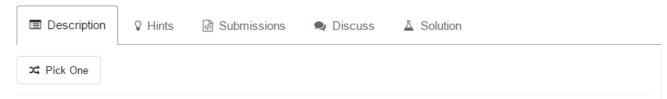
202. Happy Number



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:

```
Input: 19
Output: true
Explanation:

1<sup>2</sup> + 9<sup>2</sup> = 82
8<sup>2</sup> + 2<sup>2</sup> = 68
6<sup>2</sup> + 8<sup>2</sup> = 100
1<sup>2</sup> + 0<sup>2</sup> + 0<sup>2</sup> = 1
```

```
Seen this question in a real interview before? Yes No
```

```
public class L202 {
    public boolean isHappy(int n) {
         //用hashSet记录下来出现的数字,看是不是陷入循环中,如果存在,就代表陷入循环了
         HashSet<Integer> hashSet = new HashSet<>();
         hashSet.add(n);
         while(n != 1) {
             String tmp = new String(n+"");
             int sum = 0;
             for(int i = 0; i < tmp.length(); i ++) {</pre>
                 char c = tmp.charAt(i);
                 sum += Math.pow(c - '0', 2);
             //陷入循环,代表不是happy numbers
             if(hashSet.contains(sum)) {
                 return false;
             }else {
                hashSet.add(sum);
             n = sum;
         return true;
     }
     public static void main(String [] args) {
         System.out.println(new L202().isHappy(19));
     }
```