

## 202 快乐数

Label: 数学

编写一个算法来判断一个数  $n$  是不是快乐数。

「快乐数」定义为：

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  $1$ ，也可能是 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。

如果  $n$  是快乐数就返回 `true`；不是，则返回 `false`。

输入: 19

输出: `true`

解释:

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

$1^2 + 0^2 + 0^2 = 1$

输入:  $n = 2$

输出: `false`

- Hash判环

```
class Solution {
    private int getNext(int n) {
        int totalSum = 0;
        while (n > 0) {
            int d = n % 10;
            n = n / 10;
            totalSum += d * d;
        }
        return totalSum;
    }

    public boolean isHappy(int n) {
        Set<Integer> seen = new HashSet<>();
        while (n != 1 && !seen.contains(n)) { // 如果出现包含，则证明循环
            seen.add(n);
            n = getNext(n);
        }
        return n == 1;
    }
}
```

- 快慢指针判环

```
class Solution {
    public int getNext(int n) {
        int totalSum = 0;
        while (n > 0) {
            int d = n % 10;
            n = n / 10;
            totalSum += d * d;
        }
        return totalSum;
    }

    public boolean isHappy(int n) {
        int slowRunner = n;
        int fastRunner = getNext(n);
        while (fastRunner != 1 && slowRunner != fastRunner) {
            slowRunner = getNext(slowRunner);
            fastRunner = getNext(getNext(fastRunner));
        }
        return fastRunner == 1;
    }
}
```

- 硬编码

```
class Solution {
    private static Set<Integer> cycleMembers =
        new HashSet<>(Arrays.asList(4, 16, 37, 58, 89, 145, 42, 20));

    public int getNext(int n) {
        int totalSum = 0;
        while (n > 0) {
            int d = n % 10;
            n = n / 10;
            totalSum += d * d;
        }
        return totalSum;
    }

    public boolean isHappy(int n) {
        while (n != 1 && !cycleMembers.contains(n)) {
            n = getNext(n);
        }
        return n == 1;
    }
}
```