

0202. 快乐数

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：哈希表、数学、双指针
- 难度：简单

题目链接

- [0202. 快乐数 - 力扣](#)

题目大意

描述： 给定一个整数 n 。

要求： 判断 n 是否为快乐数。

说明：

- 快乐数定义：
 - 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。
- $1 \leq n \leq 2^{31} - 1$ 。

示例：

- 示例 1:

输入: $n = 19$

输出: `True`

解释:

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

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

py

- 示例 2:

输入: `n = 2`

输出: `False`

py

解题思路

思路 1: 哈希表 / 集合

根据题意, 不断重复操作, 数可能变为 1, 也可能是无限循环。无限循环其实就相当于链表形成了闭环, 可以用哈希表来存储为一位生成的数, 每次判断该数是否存在于哈希表中。如果已经出现在哈希表里, 则说明进入了无限循环, 该数就不是快乐数。如果没有出现则将该数加入到哈希表中, 进行下一次计算。不断重复这个过程, 直到形成闭环或者变为 1。

思路 1: 代码

```
class Solution:
    def getNext(self, n: int):
        total_sum = 0
        while n > 0:
            n, digit = divmod(n, 10)
            total_sum += digit ** 2
        return total_sum

    def isHappy(self, n: int) -> bool:
        num_set = set()
        while n != 1 and n not in num_set:
            num_set.add(n)
            n = self.getNext(n)
        return n == 1
```

py

思路 1: 复杂度分析

- 时间复杂度: $O(\log n)$ 。

- 空间复杂度: $O(\log n)$ 。