

Problem 202: Happy Number

Problem Information

Difficulty: Easy

Acceptance Rate: 58.84%

Paid Only: No

Tags: Hash Table, Math, Two Pointers

Problem Description

Write an algorithm to determine if a number `n` 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.
- * 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.

Return `true` _if_ `n` _is a happy number, and_ `false` _if not_.

Example 1:

Input: n = 19 **Output:** true **Explanation:** $1^2 + 9^2 = 82$ $8^2 + 2^2 = 68$ $6^2 + 8^2 = 100$ $1^2 + 0^2 + 0^2 = 1$

Example 2:

Input: n = 2 **Output:** false

Constraints:

$1 \leq n \leq 2^{31} - 1$

Code Snippets

C++:

```
class Solution {  
public:  
    bool isHappy(int n) {  
  
    }  
};
```

Java:

```
class Solution {  
    public boolean isHappy(int n) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def isHappy(self, n: int) -> bool:
```