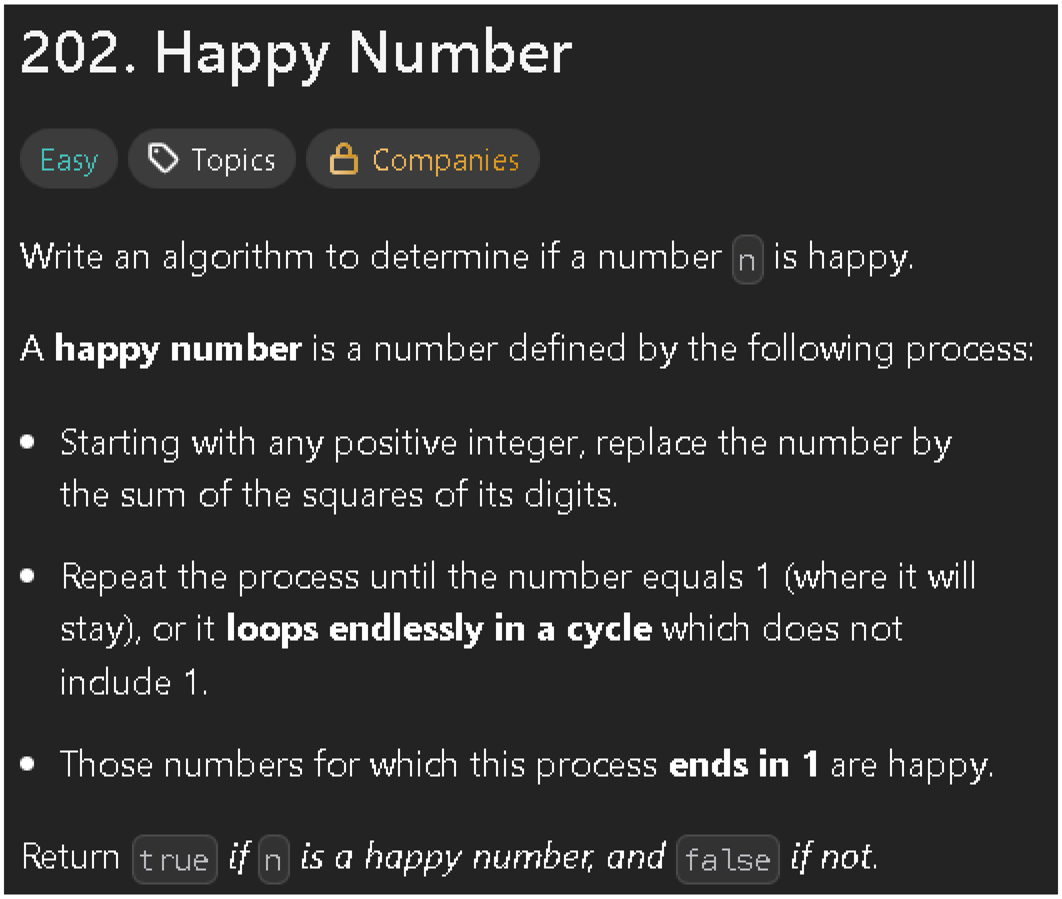


202. Happy Number

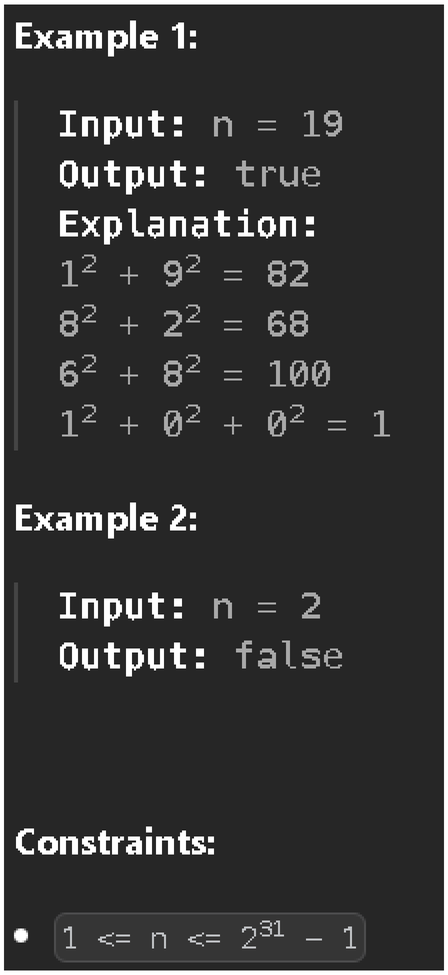
Easy Topics Companies

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:

- $1 <= n <= 2^{31} - 1$

Python:

```
class Solution:
    def isHappy(self, n: int) -> bool:
        def get_next(num: int) -> int:
            total_sum = 0
            while num > 0:
                digit = num % 10
                total_sum += digit * digit
                num //= 10
            return total_sum

        seen = set()
        while n != 1 and n not in seen:
            seen.add(n)
            n = get_next(n)

        return n == 1
```

JavaScript:

```
/**
 * @param {number} n
 * @return {boolean}
 */
var isHappy = function(n) {
    let seen = new Set();

    // Helper function: calculate sum of squares of digits
    function getNext(num) {
        let sum = 0;
        while (num > 0) {
            let digit = num % 10;
            sum += digit * digit;
            num = Math.floor(num / 10);
        }
        return sum;
    }

    while (n !== 1 && !seen.has(n)) {
        seen.add(n);
        n = getNext(n);
    }

    return n === 1;
};
```

Java:

```
import java.util.HashSet;

class Solution {
    public boolean isHappy(int n) {
        HashSet<Integer> seen = new HashSet<>();

        while (n != 1 && !seen.contains(n)) {
            seen.add(n);
            n = getNext(n);
        }

        return n == 1;
    }

    private int getNext(int n) {
        int sum = 0;
        while (n > 0) {
            int digit = n % 10;
            sum += digit * digit;
            n /= 10;
        }
        return sum;
    }
}
```