Problem 788: Rotated Digits

Problem Information

Difficulty: Medium
Acceptance Rate: 56.66%
Paid Only: No
Tags: Math, Dynamic Programming

Problem Description

An integer `x` is a **good** if after rotating each digit individually by 180 degrees, we get a valid number that is different from `x`. Each digit must be rotated - we cannot choose to leave it alone.

A number is valid if each digit remains a digit after rotation. For example:

* `0`, `1`, and `8` rotate to themselves, * `2` and `5` rotate to each other (in this case they are rotated in a different direction, in other words, `2` or `5` gets mirrored), * `6` and `9` rotate to each other, and * the rest of the numbers do not rotate to any other number and become invalid.

Given an integer `n`, return _the number of**good** integers in the range _`[1, n]`.

Example 1:

Input: n = 10 **Output:** 4 **Explanation:** There are four good numbers in the range [1, 10] : 2, 5, 6, 9. Note that 1 and 10 are not good numbers, since they remain unchanged after rotating.

Example 2:

Input: n = 1 **Output:** 0

Example 3:

Input: n = 2 **Output:** 1

Constraints:

* `1 <= n <= 104`

Code Snippets

C++:

```
class Solution {
public:
int rotatedDigits(int n) {

}
};
```

Java:

```
class Solution {
public int rotatedDigits(int n) {

}
}
```

Python3:

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