Problem 3581: Count Odd Letters from Number

Problem Information

Difficulty: Easy
Acceptance Rate: 84.65%
Paid Only: Yes
Tags: Hash Table, String, Simulation, Counting

Problem Description

You are given an integer `n` perform the following steps:

* Convert each digit of `n` into its _lowercase English word_ (e.g., 4 -> "four", 1 -> "one"). * **Concatenate** those words in the **original digit order** to form a string `s`.

Return the number of **distinct** characters in `s` that appear an **odd** number of times.

Example 1:

Input: n = 41

Output: 5

Explanation:

41 -> `"fourone"`

Characters with odd frequencies: `'f'`, `'u'`, `'r'`, `'n'`, `'e'`. Thus, the answer is 5.

Example 2:

Input: n = 20

Output: 5

Explanation:

20 -> `"twozero"`

Characters with odd frequencies: `'t'`, `'w'`, `'z'`, `'e'`, `'r'`. Thus, the answer is 5.

Constraints:

* `1 <= n <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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