


yorkshire
★ 684
June 2, 2019 5:47 AM
121 VIEWS

- 11
- Convert N to a string.
Map each digit of N to its rotation, returning False if no rotation.
Compare the reversed mapping to the original string.

```
def confusingNumber(self, N):
    S = str(N)
    rotation = {"0": "0", "1": "1", "6": "9", "8": "8", "9": "6"}
    result = []

    for c in S[::-1]: # iterate in reverse
        if c not in rotation:
            return False
        result.append(rotation[c])

    return "".join(result) != S
```

Comments: 2

[Sort](#)
[Most Votes](#)
[Newest to Oldest](#)
[Oldest to Newest](#)

Type comment here... (Markdown is supported)

Post


yilongwang
★ 41
Last Edit: December 5, 2019 12:49 PM

No extra space needed:

```
def confusingNumber(self, N: int) -> bool:
    d = {'0': '0', '1': '1', '6': '9', '8': '8', '9': '6'}
    s = str(N)

    for c in s:
        if c not in d:
            return False

    for i in range(0, len(s)-1):
        if d[s[i]] != s[i+1]:
            return False

    return True
```

Read More

2
 [Reply](#)


Xyzzy123
★ 51
April 9, 2020 1:09 AM

Here's a similar approach. It builds up the numerical value using a dictionary and then compares that value to N.

```
def confusingNumber(self, N: int) -> bool:
    rotation = {"0": 0, "1": 1, "6": 9, "8": 8, "9": 6}
    num = 0

    for i, ch in enumerate(str(N)):
        if ch not in rotation:
            return False
        num = num * 10 + rotation[ch]

    return num == N
```

Read More

1
 [Reply](#)