246. Strobogrammatic Number

Easy 397 648 Add to List Share

Given a string `num` which represents an integer, return `true` *if* `num` *is a* ***strobogrammatic number***.

A **strobogrammatic number** is a number that looks the same when rotated `180` degrees (looked at upside down).

Example 1:

```
Input: num = "69"
Output: true
```

Example 2:

```
Input: num = "88"
Output: true
```

Example 3:

```
Input: num = "962"
Output: false
```

Example 4:

```
Input: num = "1"
Output: true
```

Constraints:

- `1 <= num.length <= 50`
- `num` consists of only digits.
- `num` does not contain any leading zeros except for zero itself.

Accepted 121,652 | Submissions 257,765

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook 5 Google 2

Related Topics

Hash Table Two Pointers String

Similar Questions

Strobogrammatic Number II	Medium
Strobogrammatic Number III	Hard
Confusing Number	Easy

Java | Autocomplete

```java
class Solution {

    public boolean isStrobogrammatic(String num) {

        // Initialise a map with the five-digit rotation rules
        Map<Character, Character> rotatedDigits = new HashMap<> (
            Map.of('0', '0', '1', '1', '6', '9', '8', '8', '9', '6'));

        StringBuilder rotatedStringBuilder = new StringBuilder();

        // Remember that we want to loop backwards through the string
        for (int i = num.length() - 1; i >= 0; i--) {
            char c = num.charAt(i);
            if (!rotatedDigits.containsKey(c)) {
                return false; // This must be an invalid digit.
            }
            rotatedStringBuilder.append(rotatedDigits.get(c));
        }

        String rotatedString = rotatedStringBuilder.toString();
        return num.equals(rotatedString);
    }
}
```