2217. Find Palindrome With Fixed Length

My Submissions (/contest/weekly-contest-286/problems/find-palindrome-with-fixed-length/submissions/) Back to Contest (/contest/weekly-contest-286/)

Given an integer array `queries` and a **positive** integer `intLength`, return *an array* `answer` *where* `answer[i]` *is either the* `queries[i]`th *smallest* ***positive palindrome*** *of length* `intLength` *or* `-1` *if no such palindrome exists*.

A **palindrome** is a number that reads the same backwards and forwards. Palindromes cannot have leading zeros.

User Accepted:	2490
User Tried:	4228
Total Accepted:	2602
Total Submissions:	11727
Difficulty:	Medium

Example 1:

```
Input: queries = [1,2,3,4,5,90], intLength = 3
Output: [101,111,121,131,141,999]
Explanation:
The first few palindromes of length 3 are:
101, 111, 121, 131, 141, 151, 161, 171, 181, 191, 201, ...
The 90th palindrome of length 3 is 999.
```

Example 2:

```
Input: queries = [2,4,6], intLength = 4
Output: [1111,1331,1551]
Explanation:
The first six palindromes of length 4 are:
1001, 1111, 1221, 1331, 1441, and 1551.
```

Constraints:

- `1 <= queries.length <= 5 * 10^4`
- `1 <= queries[i] <= 10^9`
- `1 <= intLength <= 15`

Discuss (https://leetcode.com/problems/find-palindrome-with-fixed-length/discuss)

JavaScript

```
const reverse = (s) => { let res = ""; for (let i = s.length - 1; ~i; i--) { res += s[i]; } return res; };

const kthPalindrome = (queries, n) => {
    let res = [];
    for (const q of queries) {
        let x = 10 ** (n - 1 >> 1) + q - 1, s = x + '';
        res.push(s.length > n + 1 >> 1 ? -1 : s + reverse(s.slice(0, s.length - n % 2)));
    }
    return res;
};
```

Use Example Testcases

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/668101688/) More Details (/submissions/detail/668101688/)

Share your acceptance!

1