5802. Count Good Numbers

My Submissions (/contest/weekly-contest-248/problems/count-good-numbers/submissions/)

Back to Contest (/contest/weekly-contest-248/)

A digit string is **good** if the digits **(0-indexed)** at **even** indices are **even** and the digits at **odd** indices are **prime** (`2` , `3` , `5` , or `7`).

- For example, `"2582"` is good because the digits (`2` and `8`) at even positions are even and the digits (`5` and `2`) at odd positions are prime. However, `"3245"` is **not** good because `3` is at an even index but is not even.

Given an integer `n` , return *the **total** number of good digit strings of length* `n` . Since the answer may be large, **return it modulo** $10^9 + 7$.

A **digit string** is a string consisting of digits `0` through `9` that may contain leading zeros.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: n = 1
Output: 5
Explanation: The good numbers of length 1 are "0", "2", "4", "6", "8".
```

Example 2:

```
Input: n = 4
Output: 400
```

Example 3:

```
Input: n = 50
Output: 564908303
```

Constraints:

- $1 <= n <= 10^{15}$

JavaScript

```
const ll = BigInt;
const int = parseInt;
const mod = ll(1e9 + 7);
const powmod = (a, b, mod) => { let r = 1n; while (b > 0n) { if (b % 2n == 1) r = r * a % mod; b >>= 1n; a = a * a % mod; } return r; };

const countGoodNumbers = (n) => {
    let cntEven = cntOdd = int(n / 2);
    if (n % 2 == 1) cntEven++;
    // pr(cntEven, cntOdd);
    // let a = 5n ** ll(cntEven);
    // let b = 4n ** ll(cntOdd);
    let res = powmod(5n, ll(cntEven), mod) * powmod(4n, ll(cntOdd), mod);
    return Number(res % mod);
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/517055901/) More Details (/submissions/detail/517055901/)

Share your acceptance!

1