6396. Count of Integers

My Submissions (/contest/weekly-contest-348/problems/count-of-integers/submissions/) Back to Contest (/contest/weekly-contest-348/)

You are given two numeric strings `num1` and `num2` and two integers `max_sum` and `min_sum`. We denote an integer `x` to be *good* if:

- `num1 <= x <= num2`
- `min_sum <= digit_sum(x) <= max_sum`.

Return *the number of good integers*. Since the answer may be large, return it modulo $10^9 + 7$.

Note that `digit_sum(x)` denotes the sum of the digits of `x`.

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

Example 1:

```
Input: num1 = "1", num2 = "12", min_num = 1, max_num = 8
Output: 11
Explanation: There are 11 integers whose sum of digits lies between 1 and 8 are 1,2,3,4,5,6,7,8,10,11, and 12. Thus, we return
```

Example 2:

```
Input: num1 = "1", num2 = "5", min_num = 1, max_num = 5
Output: 5
Explanation: The 5 integers whose sum of digits lies between 1 and 5 are 1,2,3,4, and 5. Thus, we return 5.
```

Constraints:

- $1 \le num1 \le num2 \le 10^{22}$
- $1 \le min\_sum \le max\_sum \le 400$

JavaScript

```
const initialize4DArray = (n, m, p, q) => { let r = []; for (let i = 0; i < n; i++) { let a = []; for (let j = 0; j < m; j++) { let b = []; for (let k = 0; k < p; k++) { b.push(Array(q).fill(0)); } a.push(b); } r.push(a); } return r; };
const minus_mod = (x, y, mod) => ((x - y) % mod + mod) % mod;

const ll = BigInt, mod = 1e9 + 7;
const count = (s, t, l, r) => minus_mod(go(t, l, r), go((ll(s) - 1n).toString(), l, r), mod);

const go = (s, l, r) => {
    let n = s.length, f = initialize4DArray(n + 1, 2, 2, r + 1);
    for (let i = n; i >= 0; i--) {
        for (let isLimit = 1; isLimit >= 0; isLimit--) {
            for (let isNum = 1; isNum >= 0; isNum--) {
                for (let cnt = r; cnt >= 0; cnt--) {
                    if (i == n) {
                        f[i][isLimit][isNum][cnt] = isNum && l <= cnt && cnt <= r ? 1 : 0;
                        continue;
                    }
                    let res = 0;
                    if (!isNum) res += f[i + 1][0][0][0];
                    let L = isNum ? 0 : 1, R = isLimit ? s[i] - '0' : 9;
                    for (let digit = L; digit <= R; digit++) {
                        if (cnt + digit <= r) {
                            res += f[i + 1][(isLimit && R == digit) ? 1 : 0][1][cnt + digit];
                            res %= mod;
                        }
                    }
                    f[i][isLimit][isNum][cnt] = res % mod;
                }
            }
        }
    }
    return f[0][1][0][0];
```

```
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/963502463/) | More Details (/submissions/detail/963502463/)

Share your acceptance!