Problem 3519: Count Numbers with Non-Decreasing Digits

Problem Information

Difficulty: Hard
Acceptance Rate: 38.10%
Paid Only: No
Tags: Math, String, Dynamic Programming

Problem Description

You are given two integers, `l` and `r`, represented as strings, and an integer `b`. Return the count of integers in the inclusive range `[l, r]` whose digits are in **non-decreasing** order when represented in base `b`.

An integer is considered to have **non-decreasing** digits if, when read from left to right (from the most significant digit to the least significant digit), each digit is greater than or equal to the previous one.

Since the answer may be too large, return it **modulo** `109 + 7`.

Example 1:

Input: l = "23", r = "28", b = 8

Output: 3

Explanation:

* The numbers from 23 to 28 in base 8 are: 27, 30, 31, 32, 33, and 34. * Out of these, 27, 33, and 34 have non-decreasing digits. Hence, the output is 3.

Example 2:

Input: l = "2", r = "7", b = 2

Output: 2

Explanation:

* The numbers from 2 to 7 in base 2 are: 10, 11, 100, 101, 110, and 111. * Out of these, 11 and 111 have non-decreasing digits. Hence, the output is 2.

Constraints:

* `1 <= l.length <= r.length <= 100` * `2 <= b <= 10` * `l` and `r` consist only of digits. * The value represented by `l` is less than or equal to the value represented by `r`. * `l` and `r` do not contain leading zeros.

Code Snippets

C++:

```
class Solution {
public:
int countNumbers(string l, string r, int b) {

}
};
```

Java:

```
class Solution {
public int countNumbers(String l, String r, int b) {

}
}
```

Python3:

```
class Solution:
def countNumbers(self, l: str, r: str, b: int) -> int:
```