Problem 2478: Number of Beautiful Partitions

Problem Information

Difficulty: Hard
Acceptance Rate: 32.52%
Paid Only: No
Tags: String, Dynamic Programming, Prefix Sum

Problem Description

You are given a string `s` that consists of the digits `'1'` to `'9'` and two integers `k` and `minLength`.

A partition of `s` is called **beautiful** if:

* `s` is partitioned into `k` non-intersecting substrings. * Each substring has a length of **at least** `minLength`. * Each substring starts with a **prime** digit and ends with a **non-prime** digit. Prime digits are `'2'`, `'3'`, `'5'`, and `'7'`, and the rest of the digits are non-prime.

Return _the number of**beautiful** partitions of _`s`. Since the answer may be very large, return it **modulo** `109 + 7`.

A **substring** is a contiguous sequence of characters within a string.

Example 1:

Input: s = "23542185131", k = 3, minLength = 2 **Output:** 3 **Explanation:** There exists three ways to create a beautiful partition: "2354 | 218 | 5131" "2354 | 21851 | 31" "2354218 | 51 | 31"

Example 2:

Input: s = "23542185131", k = 3, minLength = 3 **Output:** 1 **Explanation:** There exists one way to create a beautiful partition: "2354 | 218 | 5131".

Example 3:

Input: s = "3312958", k = 3, minLength = 1 **Output:** 1 **Explanation:** There exists one way to create a beautiful partition: "331 | 29 | 58".

Constraints:

* `1 <= k, minLength <= s.length <= 1000` * `s` consists of the digits `'1'` to `'9'`.

Code Snippets

C++:

```
class Solution {
public:
int beautifulPartitions(string s, int k, int minLength) {

}
};
```

Java:

```
class Solution {
public int beautifulPartitions(String s, int k, int minLength) {

}
}
```

Python3:

```
class Solution:
def beautifulPartitions(self, s: str, k: int, minLength: int) -> int:
```