Problem 2522: Partition String Into Substrings With Values at Most K

Problem Information

Difficulty: Medium
Acceptance Rate: 47.27%
Paid Only: No
Tags: String, Dynamic Programming, Greedy

Problem Description

You are given a string `s` consisting of digits from `1` to `9` and an integer `k`.

A partition of a string `s` is called **good** if:

* Each digit of `s` is part of **exactly** one substring. * The value of each substring is less than or equal to `k`.

Return _the**minimum** number of substrings in a **good** partition of_ `s`. If no **good** partition of `s` exists, return `-1`.

Note that:

* The **value** of a string is its result when interpreted as an integer. For example, the value of `"123"` is `123` and the value of `"1"` is `1`. * A **substring** is a contiguous sequence of characters within a string.

Example 1:

Input: s = "165462", k = 60 **Output:** 4 **Explanation:** We can partition the string into substrings "16", "54", "6", and "2". Each substring has a value less than or equal to k = 60. It can be shown that we cannot partition the string into less than 4 substrings.

Example 2:

Input: s = "238182", k = 5 **Output:** -1 **Explanation:** There is no good partition for this string.

Constraints:

* `1 <= s.length <= 105` * `s[i]` is a digit from `'1'` to `'9'`. * `1 <= k <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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