Problem 1689: Partitioning Into Minimum Number Of Deci-Binary Numbers

Problem Information

Difficulty: Medium
Acceptance Rate: 88.72%
Paid Only: No
Tags: String, Greedy

Problem Description

A decimal number is called **deci-binary** if each of its digits is either `0` or `1` without any leading zeros. For example, `101` and `1100` are **deci- binary** , while `112` and `3001` are not.

Given a string `n` that represents a positive decimal integer, return _the**minimum** number of positive **deci-binary** numbers needed so that they sum up to _`n` _._

Example 1:

Input: n = "32" **Output:** 3 **Explanation:** 10 + 11 + 11 = 32

Example 2:

Input: n = "82734" **Output:** 8

Example 3:

Input: n = "27346209830709182346" **Output:** 9

Constraints:

* `1 <= n.length <= 105` * `n` consists of only digits. * `n` does not contain any leading zeros and represents a positive integer.

Code Snippets

C++:

```
class Solution {
public:
int minPartitions(string n) {

}
};
```

Java:

```
class Solution {
public int minPartitions(String n) {

}
}
```

Python3:

```
class Solution:
def minPartitions(self, n: str) -> int:
```