(/) Day 12 Explore Problems(/problemset/all/) Mock(/interview/) (/contest/) Contest Discuss(/discuss/) 🚀 December LeetCoding Challenge 🚀 (https://leetcode.com/discuss/general-discussion/655704/) Store ☆ Premium (/subscribe?ref=nb_npl)

5626. Partitioning Into Minimum Number Of Deci-Binary Numbers

My Submissions (/contest/weekly-contest-219/problems/partitioning-into-minimum-number-of-deci-binary-numbers/submissions/)

Back to Contest (/contest/weekly-contest-219/)

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

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 <= 10`^{5}
- `n` consists of only digits.
- `n` does not contain any leading zeros and represents a positive integer.

Java

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

□ **Custom Testcase** Use Example Testcases