1689. Partitioning Into Minimum Number Of Deci-Binary Numbers

27 June 2022 09:35 AM

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

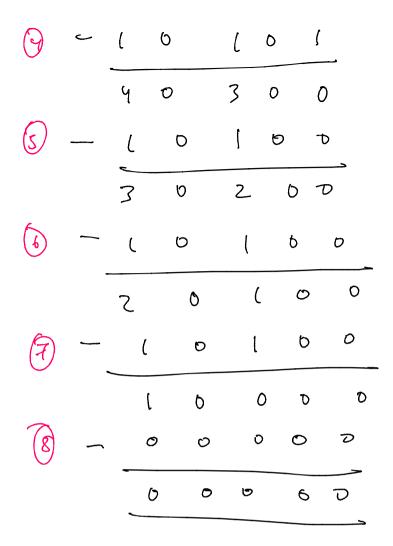
Example 3:

Input: n = "27346209830709182346"

Output: 9

$$\Lambda = 234 \Rightarrow 111 + 111 + 11 + 1 + 1 = 0(P \Rightarrow 4$$

to we will try to reduce it to zero; god is to substract max possible value (o and i)



Now it's easy to find

max_elemant (S. begin l), S. enk()) _ 0

```
    Autocomplete

 1 •
     class Solution {
     public:
2
 3 ▼
         int minPartitions(string n) {
4
              int maxi = 0;
 5 ▼
              for(auto c: n){
                  maxi = max(maxi, c - '0');
 8
9
              return maxi;
10
11
    };
```

```
i C++ ▼ • Autocomplete
  1 * class Solution {
  2
      public:
          int minPartitions(string n) {
  4 *
              int maxi = 0;
              for(auto c: n){
                  maxi = max(maxi, c - '0');
 10
              return maxi;
 11
              // Using in-built method
 13
              return *max_element(n.begin(), n.end()) - '0';
 14
 15
     };
 16
```

```
i Java

    Autocomplete

                                                                 i {} 5
       class Solution {
  1 *
  2 *
           public int minPartitions(String n) {
  3
               int maxi = 0;
  4 ▼
               for(int i = 0; i < n.length(); i++){</pre>
  5
                   maxi = Math.max(maxi, n.charAt(i) - '0');
  6
  7
               return maxi;
  8
           }
     }
  9
```