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
```

n = 2 3 4 ⇒ 111 + 111 + 11 + 1 ⇒ 4

n = 8 27 34

* we will try to reduce it to zero, goal is to substract max possible value. (0 and 1)

```
  8 2 7 3 4
- 1 1 1 1 1
---------
  7 1 6 2 3

  8 2 7 3 4
- 1 1 1 1 1
---------
  6 6 5 1 2   → Here it's '0' so substract with '0' as we cannot substract.

  8 2 7 3 4
- 1 0 1 1 1
---------
  5 0 4 0 1
```

④ — 1 0 1 0 1

4 0 3 0 0

⑤ — 1 0 1 0 0

3 0 2 0 0

⑥ — 1 0 1 0 0

2 0 1 0 0

⑦ — 1 0 1 0 0

1 0 0 0 0

⑧ — 0 0 0 0 0

0 0 0 0 0

o/p ⇒ 8

Now it's easy to find

o/p (8) 2 7 3 4 ⇒ Find max digit in a decimal number

o/p

2 7 (8) 3 4

T.C ⇒ O(D)

↳ digit

max_element(S.begin(), S.end()) — '0'

C++ Autocomplete

```cpp
class Solution {
public:
    int minPartitions(string n) {
        int maxi = 0;
        for(auto c: n){
            maxi = max(maxi, c - '0');
        }

        return maxi;
    }
};
```

C++ Autocomplete

```cpp
class Solution {
public:
    int minPartitions(string n) {
        /*
        int maxi = 0;
        for(auto c: n){
            maxi = max(maxi, c - '0');
        }

        return maxi;
        */

        // Using in-built method
        return *max_element(n.begin(), n.end()) - '0';
    }
};
```

Java Autocomplete

```java
class Solution {
    public int minPartitions(String n) {
        int maxi = 0;
        for(int i = 0; i < n.length(); i++){
            maxi = Math.max(maxi, n.charAt(i) - '0');
        }
        return maxi;
    }
}
```

Python3 Autocomplete

```python
class Solution:
    def minPartitions(self, n: str) -> int:
        return max(int(digit) for digit in n)
```