

LeetCode43_字符串相乘

原创

xyzso1z

2020-08-15 18:13:03

27

★ 收藏

编辑 版权

分类专栏: # LeetCode

文章标签:

leetcode

字符串相乘

给定两个以字符串形式表示的非负整数 `num1` 和 `num2`，返回 `num1` 和 `num2` 的乘积，它们的乘积也表示为字符串形式。

示例 1:

输入: `num1 = "2"`, `num2 = "3"`
输出: `"6"`

示例 2:

输入: `num1 = "123"`, `num2 = "456"`
输出: `"56088"`

说明:

1. `num1` 和 `num2` 的长度小于110。
2. `num1` 和 `num2` 只包含数字 0-9。
3. `num1` 和 `num2` 均不以零开头，除非是数字 0 本身。
4. 不能使用任何标准库的大数类型（比如 `BigInteger`）或直接将输入转换为整数来处理。

```
1 class Solution {
2     public String multiply(String num1, String num2) {
3         if (num1.startsWith("0") || num2.startsWith("0")) {
4             return "0";
5         }
6         int len1 = num1.length();
7         int len2 = num2.length();
8         int temp = 0;
9         int result[] = new int[len1 + len2];
10        for (int i = 0; i < len1; i++) {
11            for (int j = 0; j < len2; j++) {
12                temp = (num1.charAt(len1 - 1 - i) - '0')
13                    * (num2.charAt(len2 - 1 - j) - '0');
14                result[j + i] = result[j + i] + temp;
15            }
16        }
17        for (int i = 0; i < result.length; i++) {
18            if (result[i]/10>0) {
19                result[i+1]=result[i+1]+result[i]/10;
20            }
21            result[i]=result[i]%10;
22        }
23    }
24 }
```

```

23     StringBuffer stringBuffer = new StringBuffer();
24     boolean flag = true;
25     for (int i = result.length - 1; i >= 0; i--) {
26         if (flag && result[i] == 0) {
27             continue;
28         }
29         flag = false;
30         stringBuffer.append(result[i]);
31     }
32     return stringBuffer.toString();
33 }
34 }

```

执行用时：3 ms

内存消耗：39.7 MB

主要想法：

先计算后进位

	1	2	3	
	4	5	6	
	6	12	18	
5	10	15		
4	8	12		
↓	4	5+8	6+10+12	12+15
	4			18
	5	6	0	8
				8

<https://blog.csdn.net/xyzso1z>