

6 划分型动规

笔记本: DP Note

创建时间: 10/13/2019 3:15 PM

更新时间: 10/20/2019 3:24 PM

作者: tanziqi1756@outlook.com

Leetcode 91 Decode ways

类似Leetcode70 爬楼梯加法原理

转移方程: 对于前i个字符, 一共有前i-1种+前i-2种那么多的解码方法。

这题还有额外的难点, 就是无效字符串的处理

91. Decode Ways

难度 中等 210 收藏 分享 切换为中文 关注

题目描述

评论 (250)

题解 (61) ^{New}

提交记录

A message containing letters from A-Z is being encoded to numbers using the following mapping:

```
'A' -> 1
'B' -> 2
...
'Z' -> 26
```

Given a **non-empty** string containing only digits, determine the total number of ways to decode it.

Example 1:

Input: "12"

Output: 2

Explanation: It could be decoded as "AB" (1 2) or "L" (12).

Example 2:

Input: "226"

Output: 3

Explanation: It could be decoded as "BZ" (2 26), "VF" (22 6), or "BBF" (2 2 6).

但是这尼玛给老子输入个0是几个意思????

91. Decode Ways

难度 中等

210



收藏

分享

切换为中文

关注

题目描述

评论 (250)

题解 (61) ^{New}

提交记录

执行结果: **解答错误** [显示详情](#)

输入:

"0"

输出

1

预期结果

0

提交时间	提交结果	执行用时	内存消耗	语言
1 分钟前		N/A	N/A	Java

意思就是10表示第10个字母，这个时候10和20都会出现0

测试用例

input 00

output 0

input 10

output 1

input 01

output 0

01给老子输出0是几个意思

input 10010

output 0

应该是无效编码归零

因此，在字符串开头产生无效编码的字符串为00，01直接返回0

如果在字符串的中间出现00直接返回0

如果是这样的输入28806，直接返回0，0前面的数字大于2

这题说明了动规设定初始化的时候，还是有技巧的。

这里的初始化，一定需要N+1长度的dp数组。

91. Decode Ways

难度 中等

210



收藏

分享

切换为中文

关注

题目描述

评论 (250)

题解(61) ^{New}

提交记录

执行结果： **通过** 显示详情 >

执行用时： **1 ms**，在所有 java 提交中击败了 **99.96%** 的用户

内存消耗： **34.4 MB**，在所有 java 提交中击败了 **51.12%** 的用户

```
1 class Solution {
2     public int numDecodings(String s) {
3         int N = s.length();
4         if (N == 0) {
5             return 0;
6         }
7         // It must be N+1 but not N
8         int[] dp = new int[N+1];
9         // Initialize
10        dp[0] = 1;
11
12        for( int i = 1; i < dp.length; i++ ) {
13            dp[i] = 0;
14            if( s.charAt(i-1) != '0' ) {
15                dp[i] += dp[i-1];
16            }
17            // dp比s多一位，所以i不是i>=1,而是i>=2防止越界
18            if( i >= 2 && (s.charAt(i-2) == '1' || (s.charAt(i-2) == '2' && s.charAt(i-1) <= '6')) ) {
19                dp[i] += dp[i-2];
20            }
21        }
22        return dp[N];
23    }
24 }
```