推多米诺

n 张多米诺骨牌排成一行，将每张多米诺骨牌垂直竖立。在开始时，同时把一些多米诺骨牌向左或向右推。

每过一秒，倒向左边的多米诺骨牌会推动其左侧相邻的多米诺骨牌。同样地，倒向右边的多米诺骨牌也会推动竖立在其右侧的相邻多米诺骨牌。

如果一张垂直竖立的多米诺骨牌的两侧同时有多米诺骨牌倒下时，由于受力平衡， 该骨牌仍然保持不变。

就这个问题而言，我们会认为一张正在倒下的多米诺骨牌不会对其它正在倒下或已经倒下的多米诺骨牌施加额外的力。

给你一个字符串 dominoes 表示这一行多米诺骨牌的初始状态，其中：

- dominoes[i] = 'L'，表示第 i 张多米诺骨牌被推向左侧，
- dominoes[i] = 'R'，表示第 i 张多米诺骨牌被推向右侧，
- dominoes[i] = '.'，表示没有推动第 i 张多米诺骨牌。

返回表示最终状态的字符串。

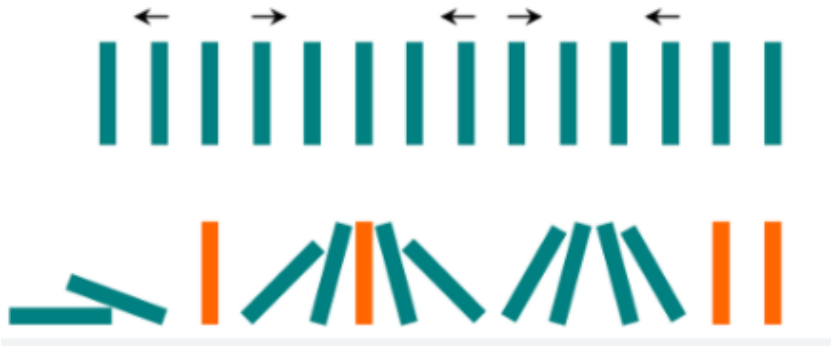
示例 1：

输入：dominoes = "RR.L"

输出："RR.L"

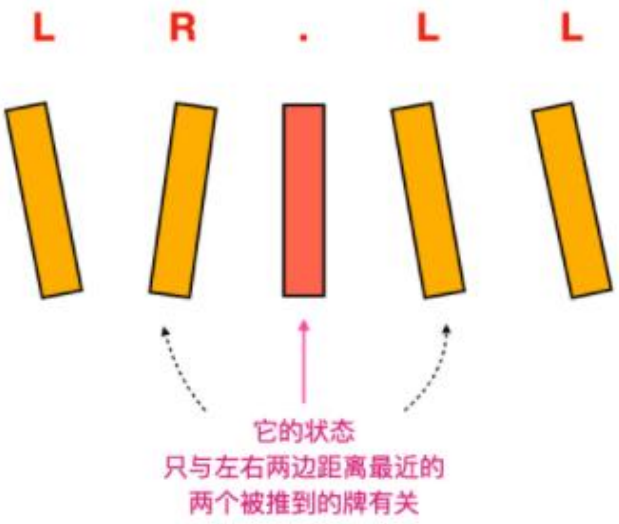
解释：第一张多米诺骨牌没有给第二张施加额外的力。

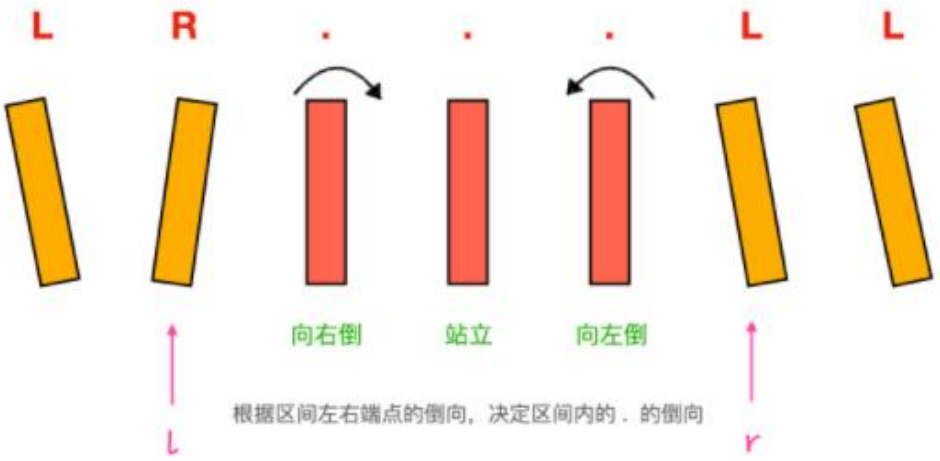
示例 2：

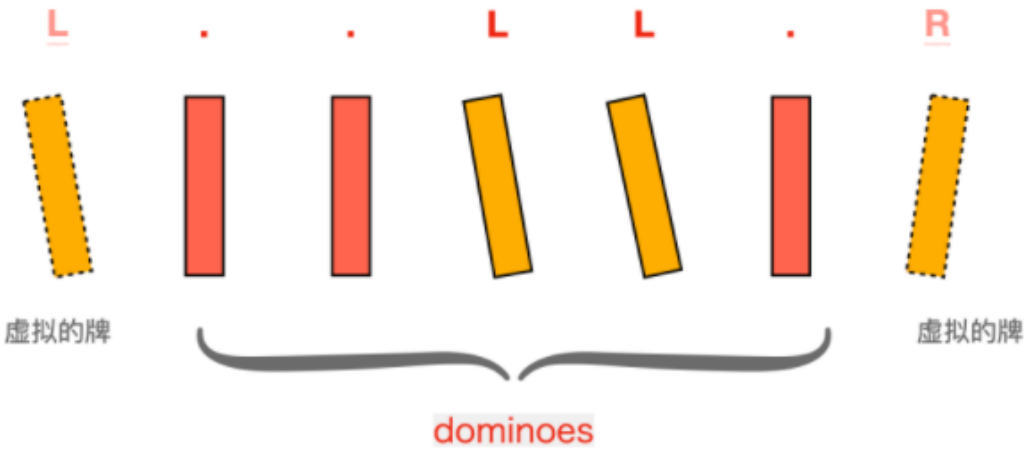


输入：dominoes = ".L.R...LR..L.."

输出："LL.RR.LLRRLL.."


L R . L L
它的状态
只与左右两边距离最近的
两个被推到的牌有关


L R . . . L L
向右倒
站立
向左倒
根据区间左右端点的倒向，决定区间内的 . 的倒向
l
r


L . . L L . R
虚拟的牌
虚拟的牌
dominoes

```python
class Solution:
    def pushDominoes(self, dominoes: str) -> str:
        '''
        'R......R' => 'RRRRRRRR'
        'R......L' => 'RRRRLLLL' or 'RRRR.LLLL'
        'L......R' => 'L......R'
        'L......L' => 'LLLLLLLL'

        '''

        dominoes = "L" + dominoes + "R"#构建虚拟左右边界 左边界向左倒 右边界向右倒 不会影响中间
        res = []
        l = 0
        for r in range(1, len(dominoes)):
            if dominoes[r] == '.':
                continue
            mid = r - l - 1 #区间长度
            if l: # 虚拟的牌不放入结果
                res.append(dominoes[l])
            if dominoes[l] == dominoes[r]:#同一方向 R......R' => 'RRRRRRRR' 或 L......L' => 'LLLLLLLL'
                res.append(dominoes[l] * mid)
            elif dominoes[l] == 'L' and dominoes[r] == 'R':#左右抵消 保持竖立 L......R' => 'L......R'
                res.append('.' * mid)
            else: #需要根据 R 和 L 数量是否为偶数来决定中间是否存在竖立情况 R......L' =>'RRRRLLLL' or 'RRRR.LLLL'
                res.append('R' * (mid // 2) + '.' * (mid % 2) + 'L' * (mid // 2))
            l = r#更新区间
        return "".join(res)
```