力扣总结版~栈

1. **最小栈**

设计一个支持 push，pop，top 操作，并能在常数时间内检索到最小元素的栈。

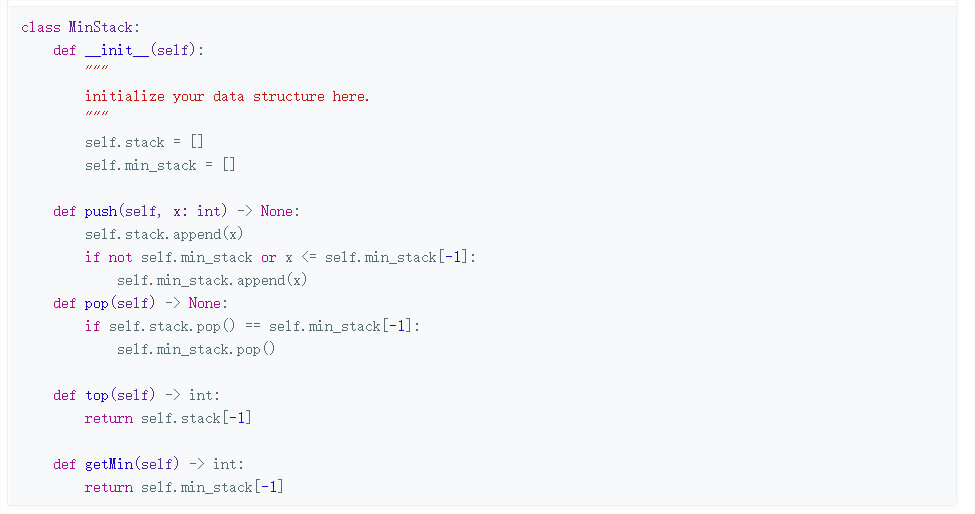
push(x) -- 将元素 x 推入栈中。

pop() -- 删除栈顶的元素。

top() -- 获取栈顶元素。

getMin() -- 检索栈中的最小元素。





1. **有效的括号**

给定一个只包括 '('，')'，'{'，'}'，'['，']' 的字符串，判断字符串是否有效。

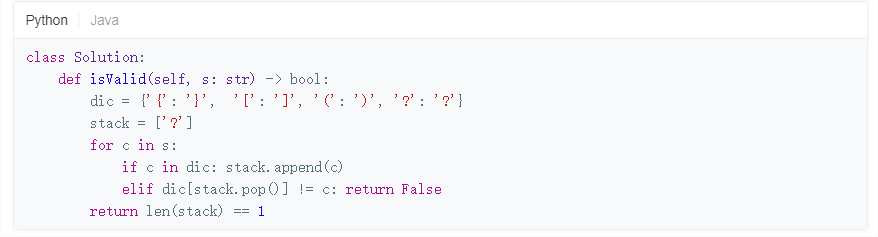
有效字符串需满足：

左括号必须用相同类型的右括号闭合。

左括号必须以正确的顺序闭合。

注意空字符串可被认为是有效字符串。





1. **字符串解码**

给定一个经过编码的字符串，返回它解码后的字符串。编码规则为: k[encoded\_string]，表示其中方括号内部的 encoded\_string 正好重复 k 次。注意 k 保证为正整数。

你可以认为输入字符串总是有效的；输入字符串中没有额外的空格，且输入的方括号总是符合格式要求的。

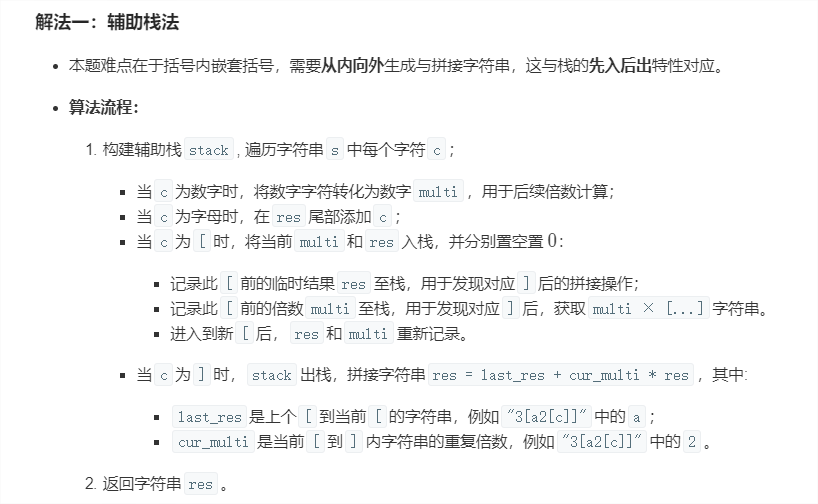
此外，你可以认为原始数据不包含数字，所有的数字只表示重复的次数 k ，例如不会出现像 3a 或 2[4] 的输入。

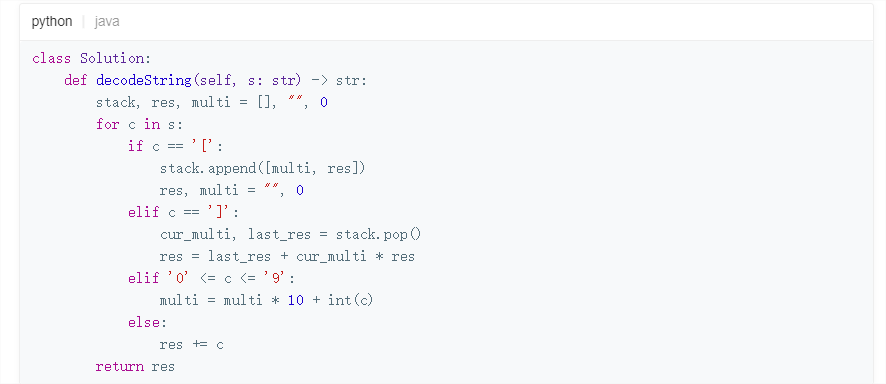
示例:

s = "3[a]2[bc]", 返回 "aaabcbc".

s = "3[a2[c]]", 返回 "accaccacc".

s = "2[abc]3[cd]ef", 返回 "abcabccdcdcdef".





1. **每日温度**

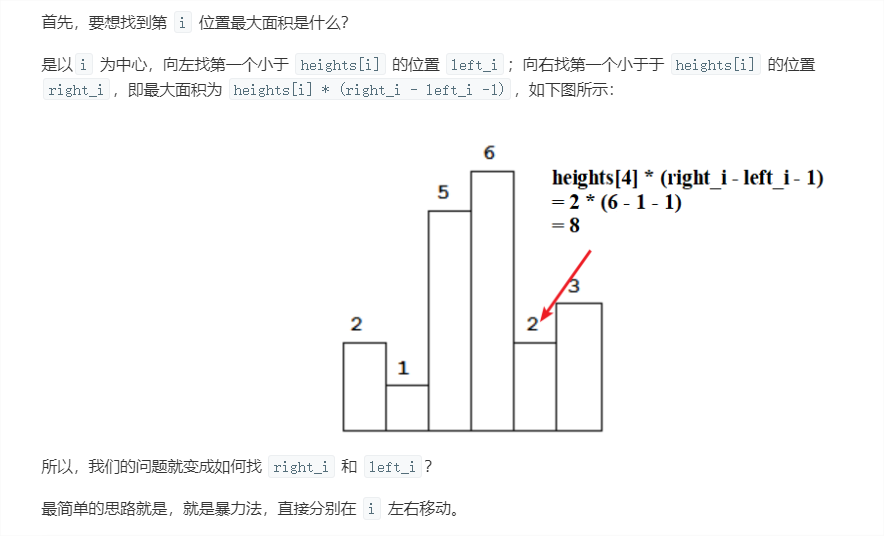
根据每日 气温 列表，请重新生成一个列表，对应位置的输入是你需要再等待多久温度才会升高超过该日的天数。如果之后都不会升高，请在该位置用 0 来代替。

例如，给定一个列表 temperatures = [73, 74, 75, 71, 69, 72, 76, 73]，你的输出应该是 [1, 1, 4, 2, 1, 1, 0, 0]。

提示：气温 列表长度的范围是 [1, 30000]。每个气温的值的均为华氏度，都是在 [30, 100] 范围内的整数。



1. **柱形图的最大矩形**





1. **最大矩形**

给定一个仅包含 0 和 1 的二维二进制矩阵，找出只包含 1 的最大矩形，并返回其面积。

示例:

输入:

[

["1","0","1","0","0"],

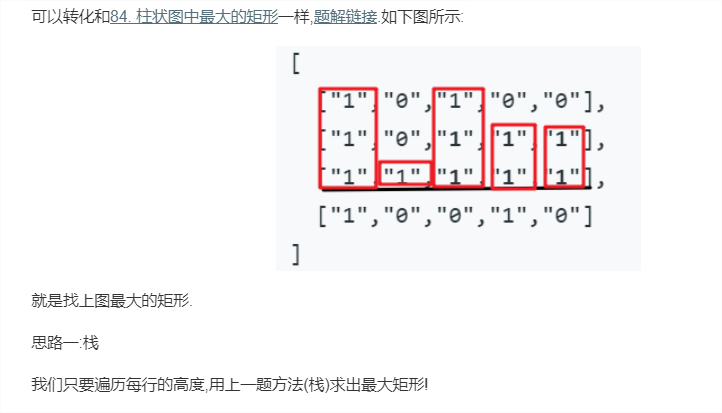
["1","0","1","1","1"],

["1","1","1","1","1"],

["1","0","0","1","0"]

]

输出: 6





1. **接雨水**

**维护递减栈**

