Assignment #4: 位操作、栈、链表、堆和 NN

Updated 1203 GMT+8 Mar 10, 2025
2025 spring, Complied by 胡新璞，工学院

1. 题目

136.只出现一次的数字

bit manipulation, https://leetcode.cn/problems/single-number/

请用位操作来实现，并且只使用常量额外空间。

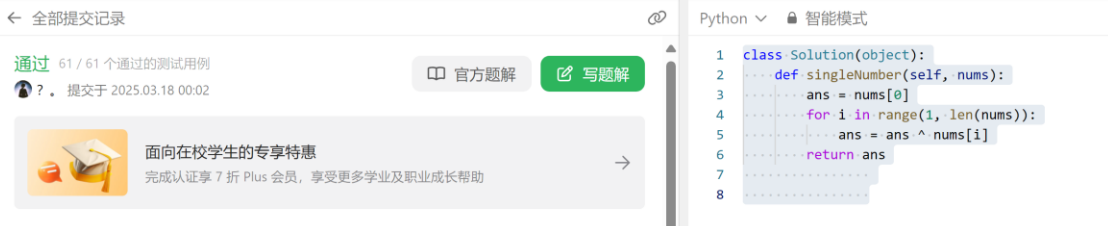
思路：一开始写出空间复杂度为 O(N)的代码，才发现这里要求了位操作，又写了一遍。

代码：

```
class Solution(object):
    def singleNumber(self, nums):
        num_set = set()
        for num in nums:
            num_set.remove(num) if num in num_set else num_set.add(num)
        return num_set.pop()

class Solution(object):
    def singleNumber(self, nums):
        ans = nums[0]
        for i in range(1, len(nums)):
            ans = ans ^ nums[i]
        return ans
```

代码运行截图 （至少包含有"Accepted"）



通过 61 / 61 个通过的测试用例

? 。 提交于 2025.03.18 00:02

面向在校学生的专享特惠

完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助

```
class Solution(object):
    def singleNumber(self, nums):
        ans = nums[0]
        for i in range(1, len(nums)):
            ans = ans ^ nums[i]
        return ans
```

20140:今日化学论文

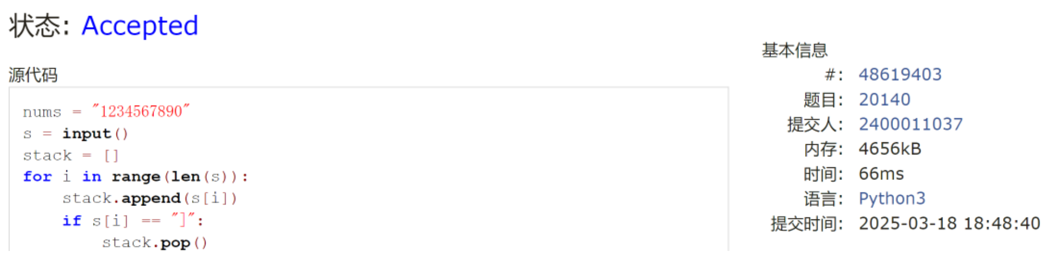
stack, http://cs101.openjudge.cn/practice/20140/

思路：用栈和辅助栈，我自己容易忽视的是把“["弹走之类的细枝末节，要多注意

代码：

```
nums = "1234567890"
s = input()
stack = []
for i in range(len(s)):
    stack.append(s[i])
    if s[i] == "]":
        stack.pop()
        stack1 = []
        while stack[-1] != "[":
            stack1.append(stack[-1])
            stack.pop()
        stack.pop()
        cnt = ""
        while stack1[-1] in nums:
            cnt = cnt + stack1[-1]
            stack1.pop()
        stack1 = stack1 * int(cnt)
        while stack1:
            stack.append(stack1[-1])
            stack1.pop()
ans = ""
for i in stack:
    ans = ans + i
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
nums = "1234567890"
s = input()
stack = []
for i in range(len(s)):
    stack.append(s[i])
    if s[i] == "]":
        stack.pop()
```

基本信息

#:	48619403
题目:	20140
提交人:	2400011037
内存:	4656kB
时间:	66ms
语言:	Python3
提交时间:	2025-03-18 18:48:40

160.相交链表

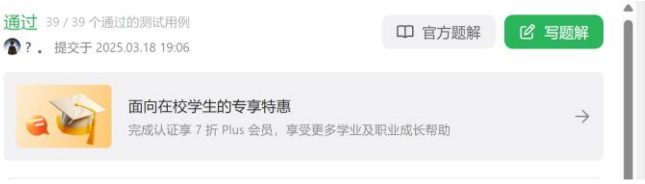
linked list, https://leetcode.cn/problems/intersection-of-two-linked-lists/

思路：题解的思路太妙了，我对链表的理解还是太浅显了

代码：

```
class Solution(object):
    def getIntersectionNode(self, headA, headB):
        a = headA
        b = headB
        while a != b:
            a = a.next if a else headB
            b = b.next if b else headA
        return a
```

代码运行截图 （至少包含有"Accepted"）

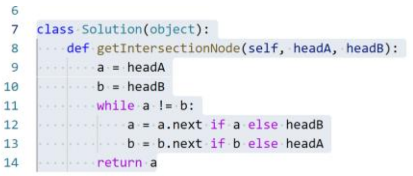


通过 39 / 39 个通过的测试用例

? 。 提交于 2025.03.18 19:06

面向在校学生的专享特惠

完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助



```
class Solution(object):
    def getIntersectionNode(self, headA, headB):
        a = headA
        b = headB
        while a != b:
            a = a.next if a else headB
            b = b.next if b else headA
        return a
```

206.反转链表

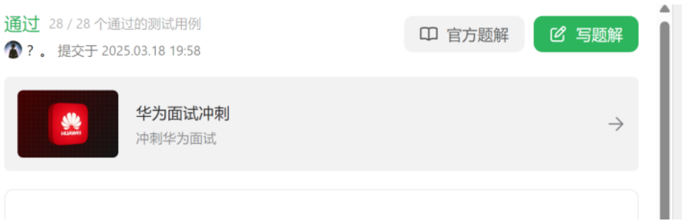
linked list, https://leetcode.cn/problems/reverse-linked-list/

思路：用迭代。本质就是类似于倒着写一遍，需要引入一个 tmp 来存中间量。递归的思路有点难懂，看了看 leetcode 上的题解试着理解了一下。

代码：

```
class Solution(object):
    def reverseList(self, head):
        cur, pre = head, None
        while cur:
            tmp = cur.next
            cur.next = pre
            pre = cur
            cur = tmp
        return pre
```

代码运行截图 （至少包含有"Accepted"）

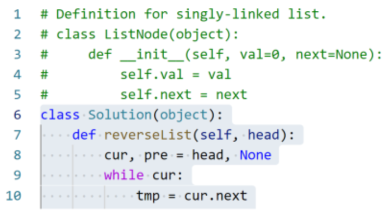


通过 28 / 28 个通过的测试用例

? 。 提交于 2025.03.18 19:58

华为面试冲刺

冲刺华为面试



```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution(object):
    def reverseList(self, head):
        cur, pre = head, None
        while cur:
            tmp = cur.next
```

3478.选出和最大的 K 个元素

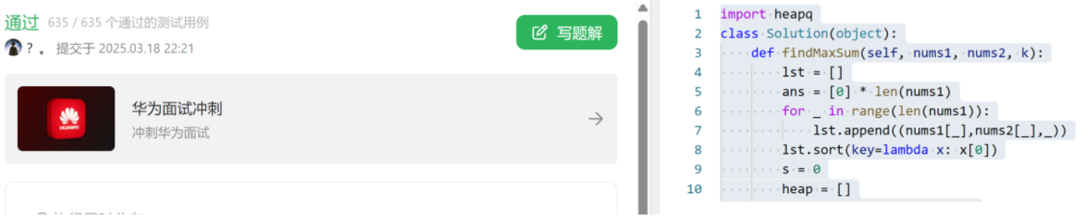
heap, https://leetcode.cn/problems/choose-k-elements-with-maximum-sum/

思路：参考了题解的想法，然后自己尝试写结果还是写了半天。。。

代码:

```
import heapq
class Solution(object):
    def findMaxSum(self, nums1, nums2, k):
        lst = []
        ans = [0] * len(nums1)
        for _ in range(len(nums1)):
            lst.append((nums1[_],nums2[_],_))
        lst.sort(key=lambda x: x[0])
        s = 0
        heap = []
        for i in range(len(lst)):
            if i >= 1 and lst[i][0] == lst[i - 1][0]:
                ans[lst[i][2]] = ans[lst[i - 1][2]]
            else:
                ans[lst[i][2]] = s
            s += lst[i][1]
            heapq.heappush(heap, lst[i][1])
            if len(heap) > k:
                s -= heapq.heappop(heap)
        return ans
```

代码运行截图 （至少包含有"Accepted"）



通过 635 / 635 个通过的测试用例

? 。 提交于 2025.03.18 22:21

华为面试冲刺

冲刺华为面试

```
import heapq
class Solution(object):
    def findMaxSum(self, nums1, nums2, k):
        lst = []
        ans = [0] * len(nums1)
        for _ in range(len(nums1)):
            lst.append((nums1[_],nums2[_],_))
        lst.sort(key=lambda x: x[0])
        s = 0
        heap = []
```

Q6.交互可视化 neural network

https://developers.google.com/machine-learning/crash-course/neural-networks/interactive-exercises

Your task: configure a neural network that can separate the orange dots from the blue dots in the diagram, achieving a loss of less than 0.2 on both the training and test data.

Instructions:

In the interactive widget:

1. Modify the neural network hyperparameters by experimenting with some of the following config settings:
   - Add or remove hidden layers by clicking the **+** and **-** buttons to the left of the **HIDDEN LAYERS** heading in the network diagram.

   - Add or remove neurons from a hidden layer by clicking the **+** and **-** buttons above a hidden-layer column.
   - Change the learning rate by choosing a new value from the **Learning rate** drop-down above the diagram.
   - Change the activation function by choosing a new value from the **Activation** drop-down above the diagram.
2. Click the Play button above the diagram to train the neural network model using the specified parameters.
3. Observe the visualization of the model fitting the data as training progresses, as well as the **Test loss** and **Training loss** values in the **Output** section.
4. If the model does not achieve loss below 0.2 on the test and training data, click reset, and repeat steps 1–3 with a different set of configuration settings. Repeat this process until you achieve the preferred results.

给出满足约束条件的截图，并说明学习到的概念和原理。

2. 学习总结和收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

正在恶补进度中，寒假偷懒的苦留到现在有好果子吃。。。独立完成了前两题，三四题看了题解的思路，然后自己写了出来（leetcode上大佬的讲解很实用，可视化做得也很清楚)，第五题折磨了好久，还是对heap不熟练，还会犯各种缩进、取值范围上的低级错误，需要加以注意，希望早点找回做题手感。