Assignment #4: 位操作、栈、链表、堆和NN

Updated 1203 GMT+8 Mar 10, 2025

2025 spring, Complied by 同学的姓名、院系

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
>
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

136.只出现一次的数字

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

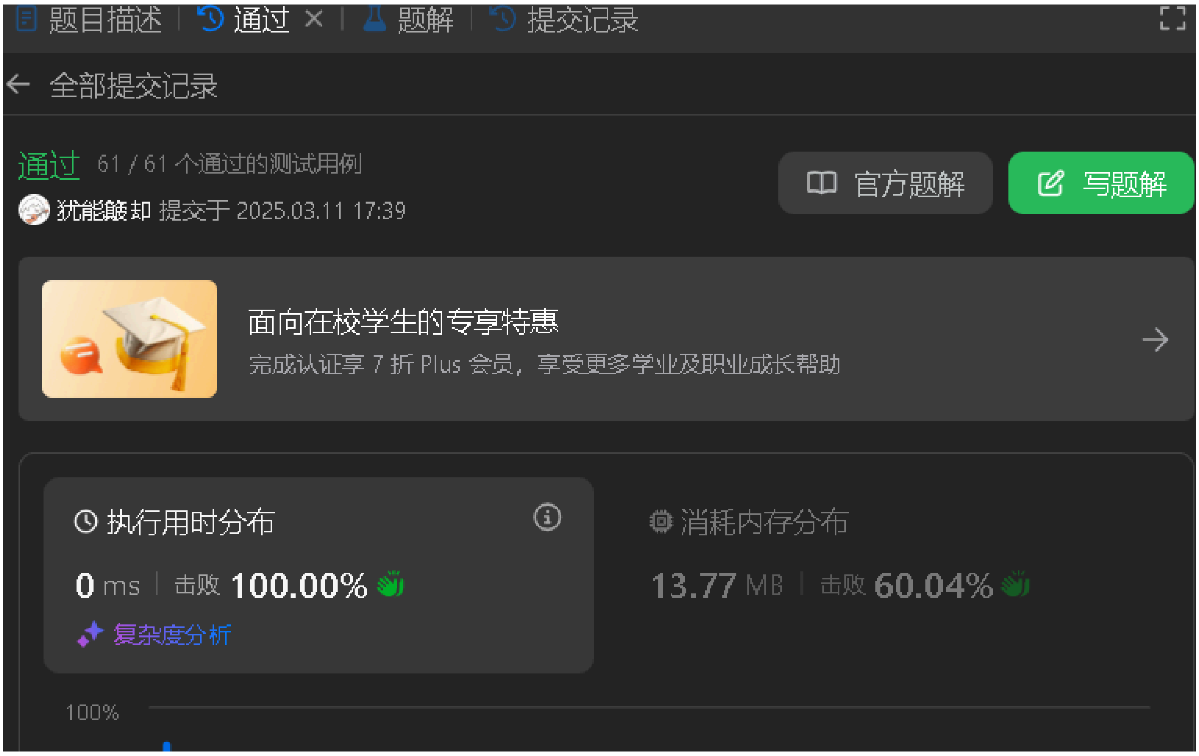
请用位操作来实现，并且只使用常量额外空间。

题目很简单，但没怎么用过位操作所以算是重新学习了一下

代码：

```
class Solution(object):
    def singleNumber(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        '''
        dic={}
        for i in nums:
            if i in dic:
                dic.pop(i)
            else:
                dic[i]=1
        key=list(dic.keys())
        return key[0]
        '''
        ###位操作
        ans=nums[0]
        for i in nums[1:]:
            ans^=i
        return ans
```

代码运行截图 （至少包含有"Accepted"）



题目描述 | 通过 × | 题解 | 提交记录

← 全部提交记录

通过 61 / 61 个通过的测试用例

犹能簸却 提交于 2025.03.11 17:39

官方题解 写题解

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

执行用时分布

0 ms | 击败 100.00%

复杂度分析

消耗内存分布

13.77 MB | 击败 60.04%

100%

20140:今日化学论文

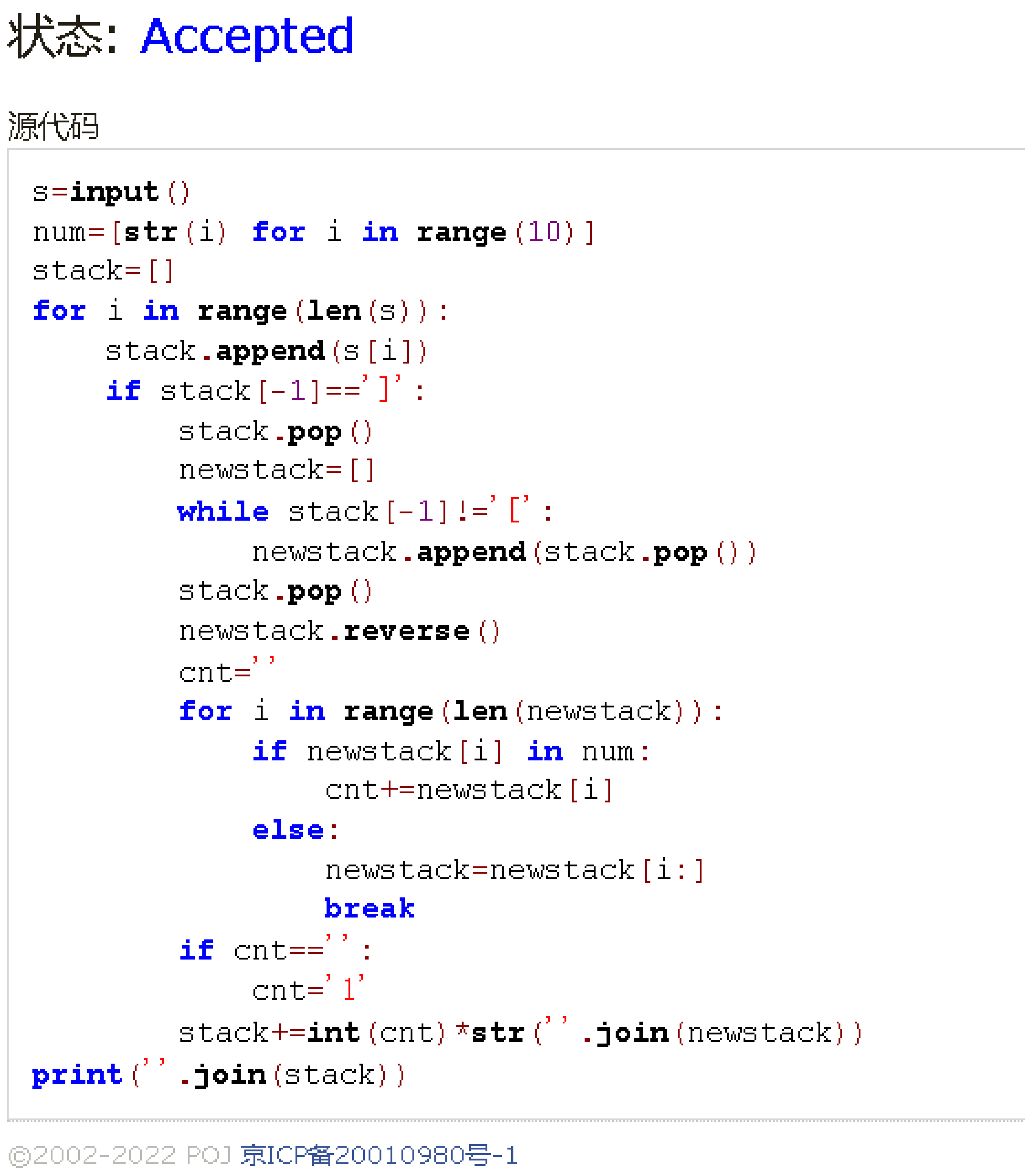
stack, http://cs101.openjudge.cn/practice/20140/

思路：

代码：

```
s=input()
num=[str(i) for i in range(10)]
stack=[]
for i in range(len(s)):
    stack.append(s[i])
    if stack[-1]==']':
        stack.pop()
        newstack=[]
        while stack[-1]!='[':
            newstack.append(stack.pop())
        stack.pop()
        newstack.reverse()
        cnt=''
        for i in range(len(newstack)):
            if newstack[i] in num:
                cnt+=newstack[i]
            else:
                newstack=newstack[i:]
                break
        if cnt=='':
            cnt='1'
        stack+=int(cnt)*str(''.join(newstack))
print(''.join(stack))
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
s=input()
num=[str(i) for i in range(10)]
stack=[]
for i in range(len(s)):
    stack.append(s[i])
    if stack[-1]==']':
        stack.pop()
        newstack=[]
        while stack[-1]!='[':
            newstack.append(stack.pop())
        stack.pop()
        newstack.reverse()
        cnt=''
        for i in range(len(newstack)):
            if newstack[i] in num:
                cnt+=newstack[i]
            else:
                newstack=newstack[i:]
                break
        if cnt=='':
            cnt='1'
        stack+=int(cnt)*str(''.join(newstack))
print(''.join(stack))
```

©2002-2022 POJ 京ICP备20010980号-1

160.相交链表

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

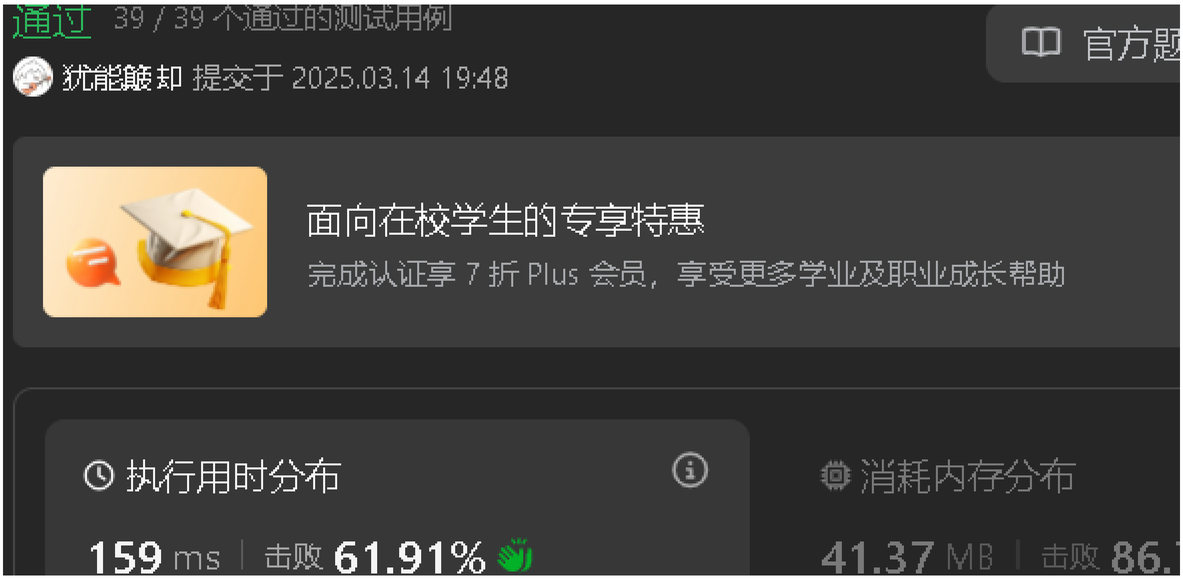
思路:

代码:

```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution(object):
    def getIntersectionNode(self, headA, headB):
        """
        :type head1, head1: ListNode
        :rtype: ListNode
        """
        if not headA or not headB:
            return None
        pointerA,pointerB=headA,headB
        while pointerA!=pointerB:
            pointerA=headB if pointerA is None else pointerA.next
            pointerB=headA if pointerB is None else pointerB.next
        return pointerA
```

代码运行截图 （至少包含有"Accepted"）



通过 39 / 39 个通过的测试用例

犹能簸却 提交于 2025.03.14 19:48

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

执行用时分布

159 ms | 击败 61.91%

消耗内存分布

41.37 MB | 击败 86.

206.反转链表

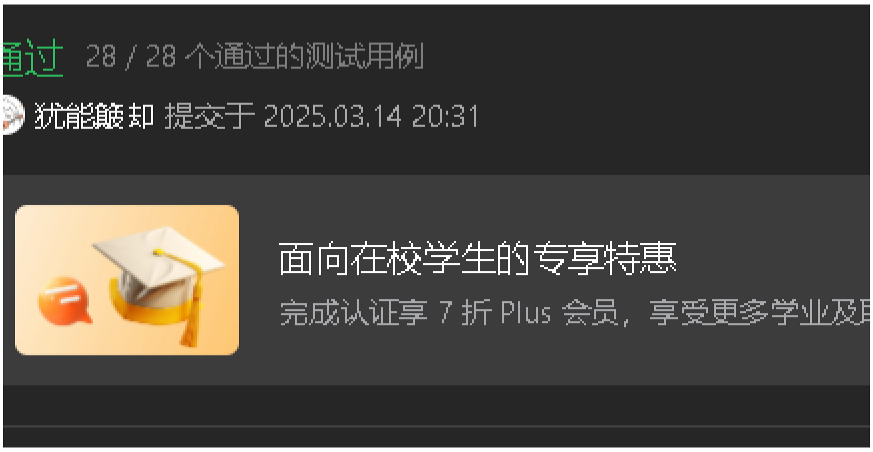
linked list, https://leetcode.cn/problems/reverse-linked-list/

思路:

代码:

```
# 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):
        """
        :type head: Optional[ListNode]
        :rtype: Optional[ListNode]
        """
        cur=head
        ans=None
        while cur:
            newnode=cur.next
            cur.next=ans
            ans=cur
            cur=newnode
        return ans
```

代码运行截图 （至少包含有"Accepted"）



通过 28 / 28 个通过的测试用例

犹能簸却 提交于 2025.03.14 20:31

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

3478.选出和最大的K个元素

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

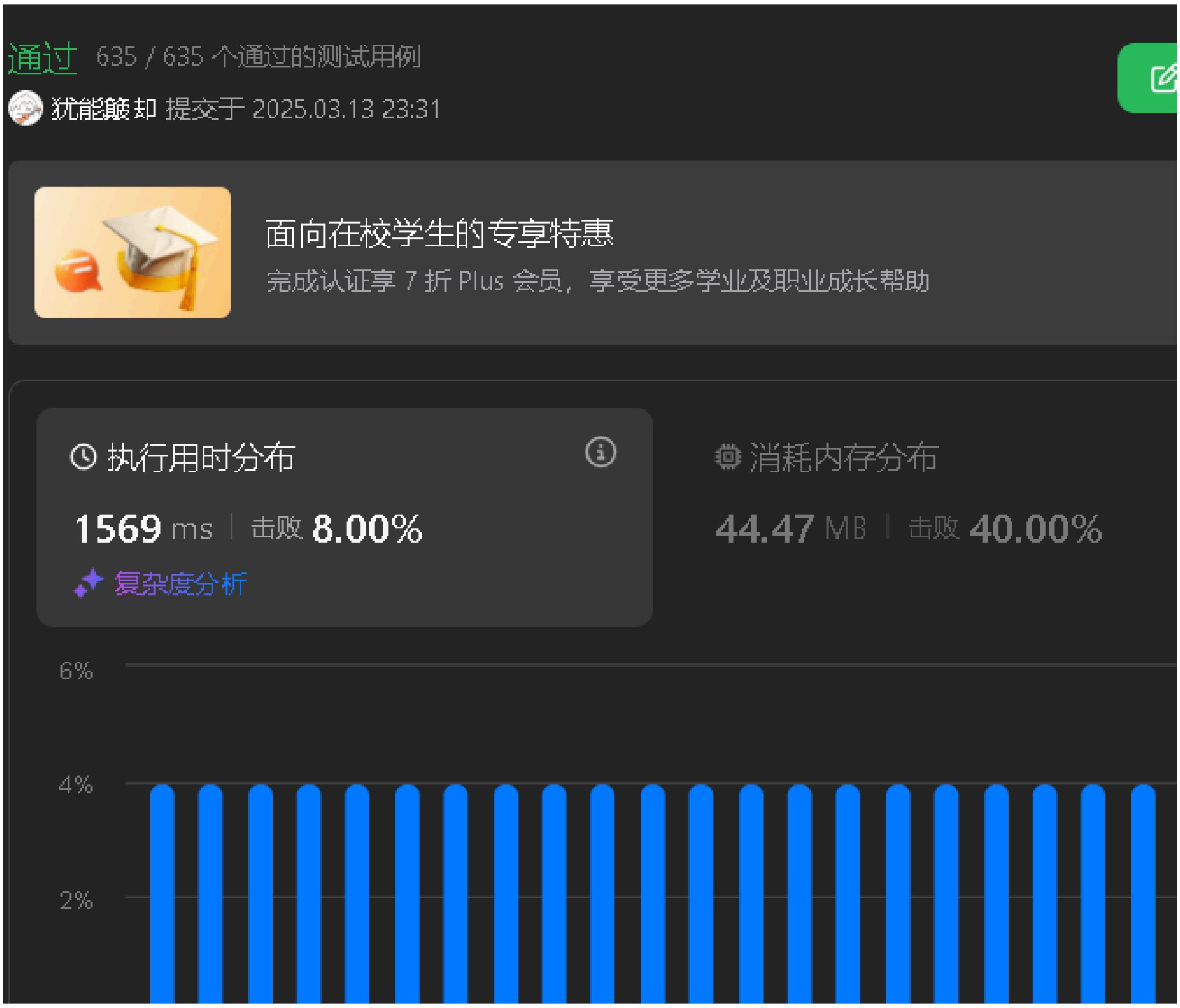
思路:

代码：

```python
import heapq
class Solution(object):
    def findMaxSum(self, nums1, nums2, k):
        """
        :type nums1: List[int]
        :type nums2: List[int]
        :type k: int
        :rtype: List[int]
        """
        lst=[(nums1[i],i,nums2[i]) for i in range(len(nums1))]
        lst.sort()
        ans=[0]*(len(nums1))
        pq=[]
        sumpq=0
        re=[]
        for i in range(1,len(nums1)):
            cnt=lst[i][1]
            if lst[i-1][0]<lst[i][0]:
                heapq.heappush(pq,lst[i-1][2])
                for a in re:
                    heapq.heappush(pq,a)
                    sumpq+=a
                    re=[]
                sumpq+=lst[i-1][2]
            else:
                re.append(lst[i-1][2])
            while len(pq)>k:
                sumpq-=heapq.heappop(pq)
            ans[cnt]=sumpq

        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

犹能簸却 提交于 2025.03.13 23:31

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

执行用时分布

1569 ms | 击败 8.00%

复杂度分析

消耗内存分布

44.47 MB | 击败 40.00%

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。

赶每日选做ing 上周没怎么做

正在看课件学习链表，这周课有事没去，OOP的用法还不太习惯啊