Assignment #4: 位操作、栈、链表、堆和NN

Updated 1203 GMT+8 Mar 10, 2025

2025 spring, Complied by 郑涵予 物理学院

> **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/

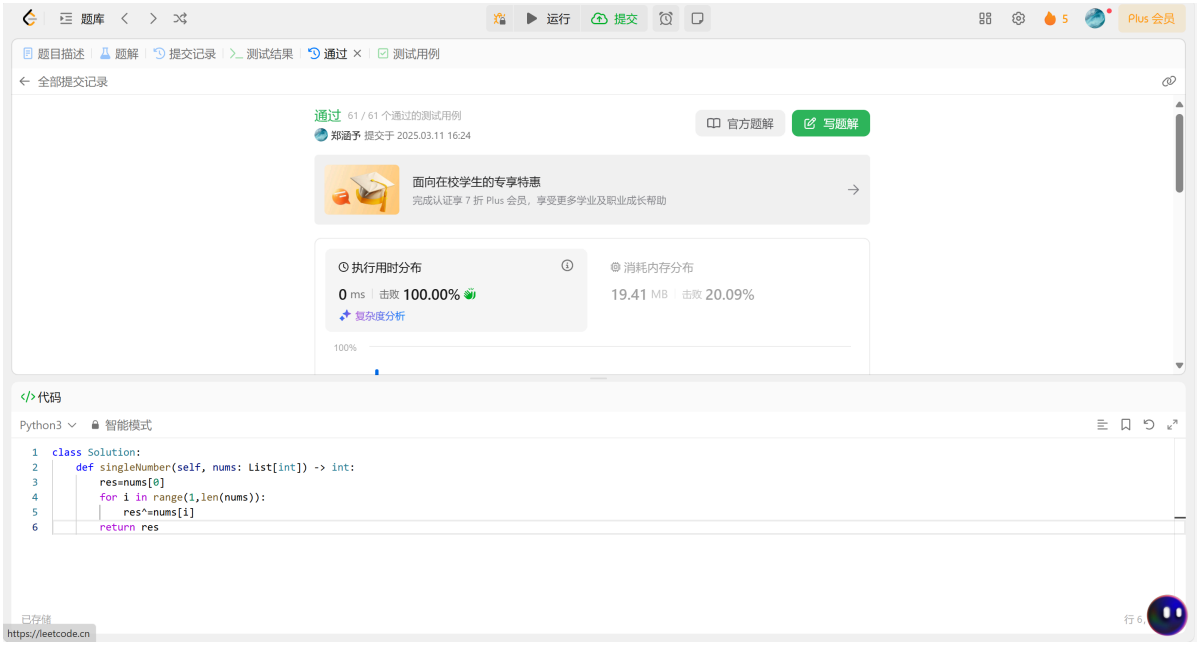
请用位操作来实现，并且只使用常量额外空间。

看到提示后能想到利用异或运算，真的是个很神奇的思路，如果没提示估计一辈子都想不出来。（用时约5min）

代码：

```python
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        res=nums[0]
        for i in range(1,len(nums)):
            res^=nums[i]
        return res
```

代码运行截图 （至少包含有"Accepted"）



20140:今日化学论文

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

思路：

利用栈进行处理，把左右括号相匹配，遇到右括号就一直pop直到左括号出现即可。（用时约10min)

代码：

```python
s=input()
num=""
stack=[]
for x in s:
    if x.isdigit():
        num+=x
    else:
        if num:
```

```
            stack.append(num)
            num=""
        if x==']':
            temp=""
            while stack and stack[-1]!='[':
                y=stack.pop()
                if y.isdigit():
                    temp*=int(y)
                else:
                    temp=y+temp
            stack.pop()
            stack.append(temp)
        else:stack.append(x)
print(''.join(stack))
```

代码运行截图 （至少包含有"Accepted"）

#48508293提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
s=input()
num=""
stack=[]
for x in s:
    if x.isdigit():
        num+=x
    else:
        if num:
            stack.append(num)
            num=""
        if x==']':
            temp=""
            while stack and stack[-1]!='[':
                y=stack.pop()
                if y.isdigit():
                    temp*=int(y)
                else:
                    temp=y+temp
            stack.pop()
            stack.append(temp)
        else:stack.append(x)
print(''.join(stack))
```

基本信息

#: 48508293
题目: 20140
提交人: 24n2400011325
内存: 3796kB
时间: 29ms
语言: Python3
提交时间: 2025-03-10 10:07:46

English 帮助 关于

160.相交链表

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

思路:

可以直接利用集合储存已经访问过的节点，遇到后直接返回。官方题解里利用双指针可以把空间复杂度降低道O（1），还是很巧妙的。（用时约5min）

代码:

```
# Definition for singly-linked list.
```

```
# class ListNode:
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution:
    def getIntersectionNode(self, headA: ListNode, headB: ListNode) ->
Optional[ListNode]:
        s=set()
        while headA:
            s.add(headA)
            headA=headA.next
        while headB:
            if headB in s:return headB
            headB=headB.next
        return None
```

代码运行截图 （至少包含有"Accepted"）

206.反转链表

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

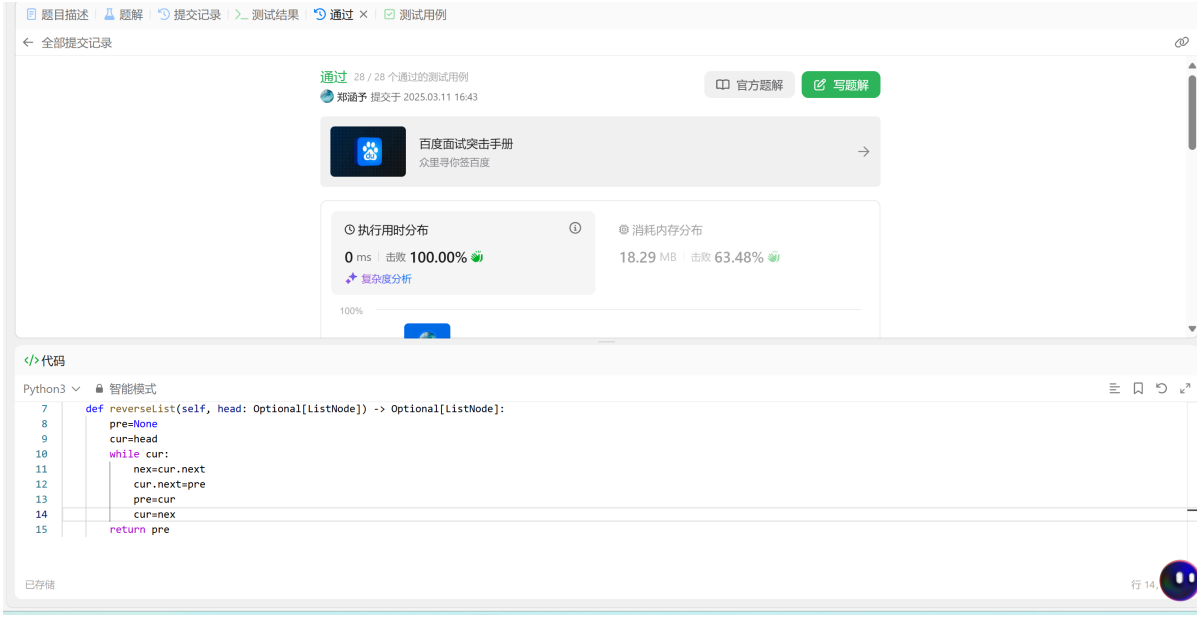
思路：

利用迭代的办法，维护前一个节点即可（用时约5min）

代码：

```python
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        pre=None
        cur=head
        while cur:
            nex=cur.next
            cur.next=pre
            pre=cur
            cur=nex
        return pre
```

代码运行截图 （至少包含有"Accepted"）



3478.选出和最大的K个元素

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

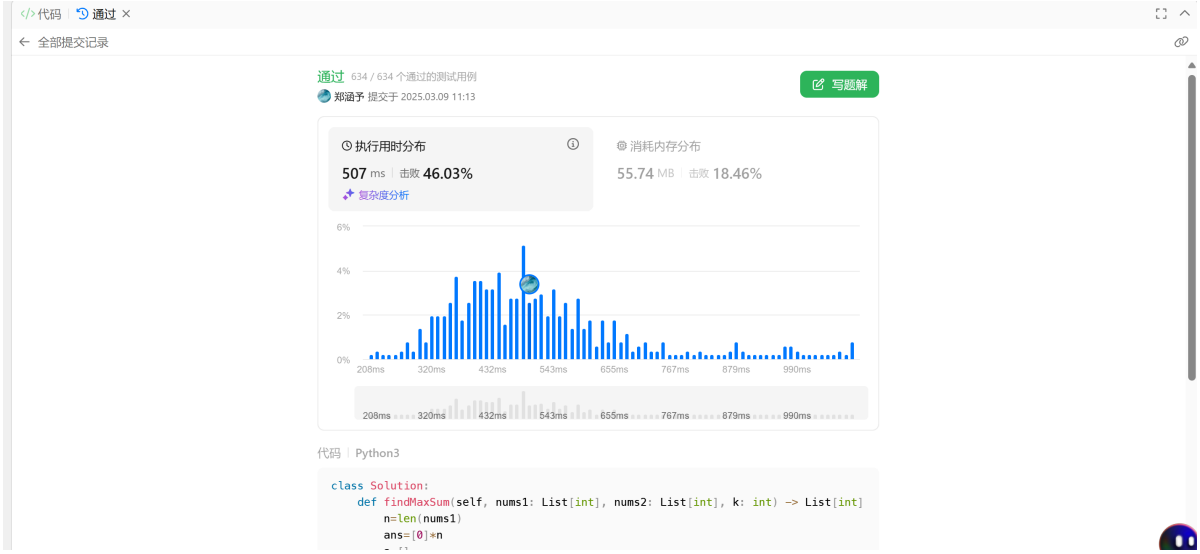
思路：

利用堆，先利用元组将对应的元素打包存到一个列表里，按照nums1[i]的值排序，其值就是对应的nums2[i]前缀最大k个，而这正好可以用heapq维护，但是要考虑到如果两个nums1[i]相同则还不能记入，所以利用一个remain数组来更新。（用时约25min)

代码：

```python
class Solution:
    def findMaxSum(self, nums1: List[int], nums2: List[int], k: int) ->
List[int]:
        n=len(nums1)
        ans=[0]*n
        a=[]
        for i in range(n):
            a.append((nums1[i],-nums2[i],i))
        a.sort()
        q=[]
        remain=deque()
        temp=0
        remain.append(a[0])
        for i in range(1,n):
            index=a[i][2]
            while remain and remain[0][0]<a[i][0]:
                t=-remain.popleft()[1]
                temp+=t
                heapq.heappush(q,t)
            while len(q)>k:
                temp-=heapq.heappop(q)
            ans[index]=temp
            remain.append(a[i])
        return ans
```

代码运行截图 （至少包含有"Accepted"）



Q6.交互可视化neural network

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

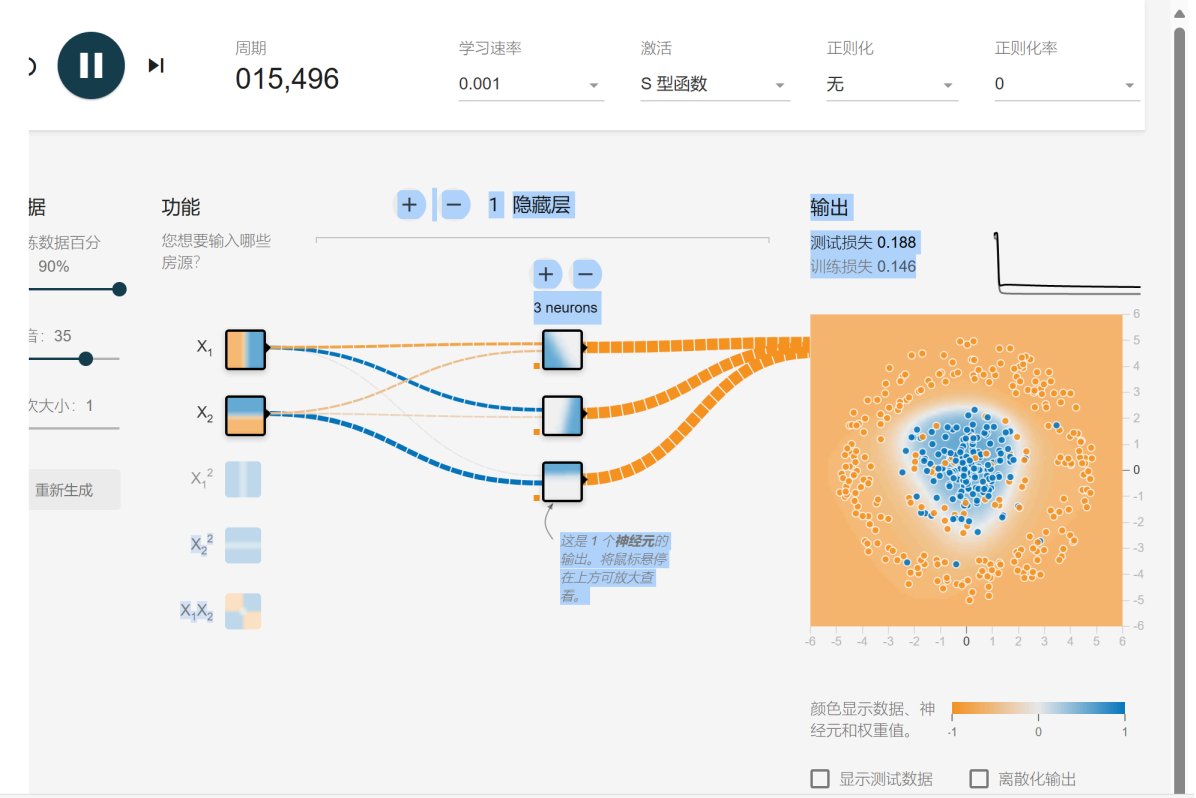
Your task: configure a neural network that can separate the orange dots from the blue dots in the diagram, achieving a loss of less than 0.2 on both the training and test data.

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。



增加了一个隐藏层，三个神经元，并且把激活调成S型函数后成功把损失降低到了0.2以下。大致了解了隐藏层，噪音，正则化等概念，不过还没来得及把前面的章节都看完。

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。

参加了力扣的周赛，这次运气不错AC了三道，感觉现在周赛第二题的难度已经和月考第五题差不多了，不过比起月考第六题还是差了点。周赛第三题用到了线段树，虽然学习过相关知识但写起来还是很不熟练，要是考试遇到估计会选择对着cheat sheet抄了。