Assignment #4: 位操作、栈、链表、堆和NN

Updated 1203 GMT+8 Mar 10, 2025

2025 spring, Complied by 叶靖、信管

> **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。

代码：

```
class Solution: def singleNumber(self, nums: List[int]) → int: ans = 0 for num in nums: ans ^= num return ans
```

代码运行截图 （至少包含有"Accepted"）

04-01

20140:今日化学论文

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

思路:

代码:

string = input() ans = []

for i in range(len(string)): ans.append(string[i]) if ans[-1] == ']': ans.pop() stack = [] while ans[-1] != '[': stack.append(ans.pop()) ans.pop() numstr = '' while stack[-1] in '0123456789': numstr += str(stack.pop()) stack = stack * int(numstr) while stack != []: ans.append(stack.pop()) print(*ans, sep='')

代码运行截图 （至少包含有"Accepted"）

04-02

160.相交链表

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

思路:

代码:

class Solution: def getIntersectionNode(self, headA: ListNode, headB: ListNode) → Optional[ListNode]: if not headA or not headB: return None p1, p2 = headA, headB while p1 != p2: p1 = p1.next if p1 else headB p2 = p2.next if p2 else headA return p1

代码运行截图 （至少包含有"Accepted"）

04-03

206.反转链表

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

思路:

迭代法，双指针

代码:

class Solution: def reverseList(self, head: Optional[ListNode]) → Optional[ListNode]: prev = None curr = head

```
    while curr:
        next_node = curr.next
        curr.next = prev
        prev = curr
        curr = next_node
    return prev
```

代码运行截图 （至少包含有"Accepted"）

04-04

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：

import heapq

class Solution: def findMaxSum(self, nums1: List[int], nums2: List[int], k: int) → List[int]: n = len(nums1) result = [0] * n pairs = sorted((num, i) for i, num in enumerate(nums1)) min_heap = [] total_sum = 0 j = 0

```
for value, i in pairs:
    while j < n and pairs[j][0] < value:
        _, idx = pairs[j]
        heapq.heappush(min_heap, nums2[idx])
        total_sum += nums2[idx]
        if len(min_heap) > k:
            total_sum -= heapq.heappop(min_heap)
        j += 1
    result[i] = total_sum
return result
```

代码运行截图 （至少包含有"Accepted"）

04-05

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。

可能是练习做多了，开始有手感了，所以解题的速度开始提升了。这份功课最大的问题出现在leetcode上，因为用惯了pycharm，需要调整代码写法来满足leetcode给定的环境。