Week1 学习心得

1. 我学到了什么知识？
2. 理论知识及代码实现数据结构方面：

课程的学习中，学习了解数组、链表、队列、栈等四种数据结构及他们的时间空间复杂度的分析，学习怎么用Python来实现这四种数据结构及相应的接口等。学会怎么去查询相关的源代码文档及相关信息，逼迫自己去看去学习源代码。

存在的问题：运用Python实现这些数据结构的熟练程度依然不够，尤其在链表的部分，链表的反转，循环链表等常常会发生指针丢失的情况，仍然需要进一步的练习。

对照自己原先绘制的数据结构脑图，在受限线性表中，越界错误的知识点及静态链表的知识点，仍然没有掌握。

图片包含 屏幕截图

描述已自动生成

图1 非受限线性表

图片包含 屏幕截图

描述已自动生成

图2 受限线性表（栈&队列）

在学习的过程中，Python的数组array跟numpy的array一直在困扰着我，疑惑我们写算法的时候，能直接调用第三方库numpy的array吗？以及array跟numpy的array存在什么差异，运用的时候选择哪个为优，这是我后续还要探讨的问题。

1. 算法题知识点总结：
2. 双指针法，算法题26删除重复数值及283删除零，我认为是两道类似的题，可以纳入到一类中，他们用一个指针帮助记录目标数值的下标，用另一个指针下标来遍历数组，使用额外的空间nums[i]来记录下一个要比较的值，额外空间为O（1）。我在这道题中常犯的错误是：常常不会借助额外的空间nums[i]记录比较数值，单纯的运用下标nums[j+1]与nums[j]对比，进而导致j循环中，无法解决指针溢出的问题
3. 链表的算法题，常常需要设置一个“哨兵“指针，或前置指针，我在链表的指针中常常迷失自我。在链表题中，一步一步画图，很关键，帮助梳理自己的思路。链表的题，死命的写，死命的重复就对了。在链表中，自己容易出现的错误是：把链表当数组，遍历链表一直要写成数组下标遍历的形式。
4. 栈的应用可以用来解决类似最近相似问题，类似“剥洋葱“的结构。左右括号匹配的问题，可以用字典结构来解决。
5. 队列，时间关系，这周队列的学习还没有完全完成。
6. 算法题实战心得
7. 需要提升自己写代码能力

在算法题的实战中，发现除了暴力解法，大部分情况下自己是想不到更优化的解法的。在暴力法求解的过程中，自己编程能力弱的缺点也一直一直在曝露出来。尤其在写遍历循环的时候，很容易出现下标溢出的问题。

（2）理解后再去背及仿写代码并且**不断重复不断重复**才能真的学会

对于算法题，现阶段只能借助题解及老师的讲解，先学习其他人的代码，模仿着写，背诵默写，让自己达到熟悉相关题型的程度。实际写题目的时候，发现背代码，理解不透彻的话，在自己做题的时候，所有的不理解的问题就都会暴露出来了。因此，每一次的重复，都是在发现自己的问题所在，是检验自己是不是真的理解，是不是真的掌握的那类题型。

（3）画图是有效的学习手段

题目理解的过程中，需要不断的画图，画图，且在画图的过程中，不要太相信自己，不要轻易的跳步。跳步的话就很容易出现下图的悲催情况…



图3 写链表过程中的我大概就是如此吧…

1. **总结**

这一周的学习，痛苦着，痛苦着，痛苦着快乐着。

1. 让我了解到自己的学习边界，清楚自己的基础有多差，进而重新来制定自己的学习目标，希望在学习结束的时候一方面能够让自己的编程能力有长足的进步，一方面让自己在数据结构与算法的学习上达到掌握老师所讲过的题目，能做出大部分的课后作业题的目标，后续还要继续锻炼与提升。
2. 这一周另一个主要的问题在于低估了算法学习的时间投入需求，没有合理的每天分配足够的时间，前两天的时间投入不够，导致后续的学习只能勉勉强强跟上，算法题的遍数过不到五遍，甚至有的连第一遍都没能完全走完。在第二周的时候，需要再增加时间的投入，计划每天固定出2小时左右进行学习。接下来在小组内进行每日学习打卡吧~
3. 不要害怕重复，不要害怕重复，不要害怕重复。在算法题中，哪怕最简单的moveZero等题目，我在周三周四的时候觉得自己已经学会了，已经会写了，等到周六正式写作业的时候，又狠狠的打击了自己。
4. 需要对每一道算法题写总结，归纳相关的题型，相似的解法。