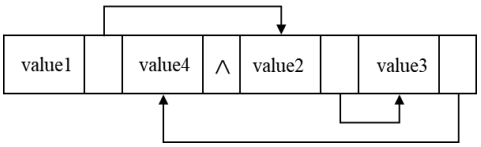
Homework 1

注意事项：

1. 书面作业需提交手写版，可以用 iPad 等设备书写
2. 请在作业首页注明姓名、学号
3. 作答时只需要写清题号，不需要抄写题目
4. 作业通过教学网提交，2025 年 3 月 11 日 15：10 截止

一、判断题（如果判断为错误，需要写出理由）

1. 同一个算法，使用不同的编程语言实现，或者在不同的机器上运行，实际的运行时间一般不同，因而时间复杂度也不同。 ✗ 时间复杂度仅与算法有关，因而相同
2. $O(n^2+100)$, $O(100n^2)$, $O(n^2 \log n)$ 表示的复杂度相同。 ✗ $O(n^2 \log n)$ 大于前两个
3. 数据结构的三要素包括逻辑结构、存储结构、以及基于数据定义的运算。顺序表、链表定义了不同的逻辑结构。 ✗ ，都是线性结构
4. 抽象数据类型（ADT）可以有多种不同的实现方案，不同方案同一操作的实现的时间复杂度也可能不同。 ✓
5. 顺序表与链表都能够以 O(1)的时间复杂度，实现对指定下标元素的访问与修改。 ✗ 顺序表 O(1)，链表 O(n)
6. 向顺序表中的任意位置插入元素，最好情况下的时间复杂度仅为 O(1)，最坏情况下的时间复杂度为 O(n)。假定不同插入位置发生的概率是均等的，则平均时间复杂度是 O(n)。 ✓
7. 当对数据的操作主要是读操作时，应该使用顺序表存储；当对数据的操作主要是插入删除时，应该使用链表存储。 ✓
8. 如果变量 p 指向双向链表中的一个结点，那么删除这个结点 p 只需要 O(1)的时间复杂度。 ✓
9. Python 中的 list 数据类型由于不具备固定的最大表长，因而存储方式并不是顺序存储。 ✗ 是顺序存储
10. 下图展示了一个线性表的实现：一段连续的内存中包含 4 个节点，使用引用（指针）链接来维护前驱与后继关系。尽管各节点在内存上连续，但该数据结构是链式存储的。 ✓



二、算法设计（每个问题只需要写明算法设计思路和必要细节即可，不需要写出所有代码，可以借助伪代码、流程图等方式表达算法流程）

1. 给定两个单链表，其中的元素都是升序存储的。设计一个算法将其合并为一个升序的单链表。分析所设计的算法的时间复杂度。

2. 给定一个单链表，设计一个算法寻找单链表的中间结点的值。若链表长度为偶数，则算法应返回中间的两个结点的前一个。分析所设计的算法的时间复杂度。

1. 双指针

(1) 创建一个虚拟头节点，使用一个指针 current 指向虚拟头节点

(2) 使用两个指针 P_1，P_2 分别连接两个链表头节点；比较 P_1 和 P_2 值大小，将较小的节点接到 current 后；移动 current 和较小节点指针

(3) 若一个链表遍历完毕，将另一个链表剩余部分连接到 current 后.

(4) 返回虚拟头节点下一个节点

时间复杂度 $O(m+n)$ ，m，n 分别为两个链表长度

2、快慢指针（双指针）

(1) 定义两个指针 slow、fast，初始时都指向链表头节点

(2) fast 每次移动两步，slow 每次移动一步

(3) 当 fast 移动到末尾（奇数长度）或 fast 无法移动两步时，慢指针所指向的节点既是单链表中间（两个中前一个）的值。

时间复杂度 O(n)，n 为链表长度