

## 作业 2

下列题目中，第 1、4、7 题和附加题需要提交代码，其余内容为纸面作业，请用 word 或 pdf 的格式进行提交。

1. 请根据第四周课件上给出的双端队列的 ADT，给出一个完整的基于数组的双端队列的 python 实现。 (20 分)
2. 请给出一个快速高效逆置单向链表的递归算法。 (10 分)
3. 仅使用固定的额外空间，不使用任何递归，给出一个逆置单向链表的算法。 (10 分)
4. 请编写一个 python 函数，将两个有序的单向链表合并为一个有序链表，合并后使原链表为空。链表的节点结构参考课件中的单向链表节点，且链表带有头节点。有序链表中元素为从小到大排列。 (20 分)
5. 一棵共有  $n$  个节点的树，其所有内部结点的度都为  $k$ ，求该树的叶子节点数。 (10 分)
6. 假设某完全二叉树共有 2023 个节点，请问：
  - a) 该二叉树有多少层？ (5 分)
  - b) 它有多少个叶子节点？ (5 分)
  - c) 它有多少个度为 2 的节点？ (5 分)
7. 如果充分利用哨兵节点 (`_sentinel`)，我们可以简化链式二叉树 `LinkedBinaryTree` 的实现过程。这里，哨兵节点为根节点的父节点，且根节点为哨兵节点的左孩子节点。请写出新的 `LinkedBinaryTree` 的 `_delete` 方法。 (15 分)

### 附加题：

增强二叉搜索树：

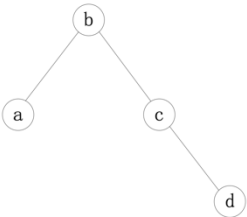
二叉搜索树有如下性质：对该树上的任意结点，若其左子树不为空，则左子树上的所有节点的关键字均小于根节点的关键字；若其右子树不为空，则右子树上的所有节点的关键字均大于等于根节点的关键字。由于其具有的结构性质，二叉搜索树常用于快速查找和排序。

增强二叉搜索树是二叉搜索树的一种推广，其中每个节点除了保存关键字外，还保存了以该结点为根的子树的结点数目（包括该结点本身）。

通常我们使用链式结构来存储二叉搜索树。在本实验中，我们要求使用链式结构实现增强二叉搜索树，并实现其建立、前/中/后序遍历、统计子树节点个数等操作。二叉搜索树的建立方式请自行翻阅教材 11.1.3 二叉搜索树的插入部分或另行查阅。可以继承课件上的 `LinkedBinaryTree` 类。

为保证构造结果的唯一性，请严格按照给定的输入顺序依次读入关键字，并使得左子树的节点关键字严格小于根节点的关键字，即遇到与根节点关键字相等的新输入，应将其插入到右子树中。

为使输出结果的表示意义唯一，对于度为 0 或 1 的节点，用 “#” 补全成度为 2 的节点，例如右图中二叉树的前序遍历关键字序列为：



b a # # c # d # #

程序需先通过输入建立二叉树，并以一定的格式输出：

输入格式：

第一行：一个正整数  $n$

第二行： $n$  个正整数，以空格隔开，代表存储在节点的关键字

输出格式：

第一行：前序遍历的节点关键字

第二行：中序遍历的节点关键字

第三行：后序遍历的节点关键字

之后程序进入查询状态，不断询问用户想要查询其节点数目的子树的根节点的关键字，如不存在此关键字则返回 0，直到用户输入 “q” 或 “Q” 程序停止运行。例如，对右图的子树节点数目进行查询，输入 6 则返回 5，输入 2 则返回 3，输入 10 则返回 13，输入 15 则返回 0。

