

顺序表/数组

矩阵/二维数组

单链表（带头节点/不带头节点）

双链表

递归

栈

队列

字符串

二叉树

搜索

图

## ✧ 顺序表/数组

---

- 1 ListInsert在指定下标位置插入元素
- 2 在顺序表中查找第一个元素值等于x的元素，并返回其次序
- 3 统计顺序表中x出现的次数
- 4 设计一个高效算法，将顺序表L的所有元素逆置，要求算法的空间复杂度为 $O(1)$
- 5 从顺序表中删除其值在给定值s与t之间(包含s和t，要求 $s < t$ )的所有元素，若s或t不合理或顺序表为空，则显示出错信息并退出运行
- 6 在一个长度为n的数组nums里的所有数字都在 $0 \sim n-1$ 的范围内。数组中某些数字是重复的，但不知道有几个数字重复了，也不知道每个数字重复了几次。请找出数组中任意一个重复的数字
- 7 给定一个顺序表，将其循环右移m位
- 8 设计一个算法将小于0的整数放在前面部分，大于等于0的整数放在后面部分
- 9 给定一个有序的顺序表，设计一个算法删除顺序表中的重复的元素
- 10 给定有序顺序表A，B，将A和B合并到C中

- 11 给定有序顺序表A, B (A和B中元素不重复出现), 求A和B的交集 (用顺序表表示)
- 12 和上面一样, 求A和B的并集
- 13 给定顺序表A, B (A和B中元素不重复出现), 求A和B的交集 (用顺序表表示)
- 14 和上面一样, 求A和B的并集
- 15 两个递增的有序数组a和b分别有n和m个数, 设计一个算法求出第k大的数
- 16 若用顺序表存储一元多项式系数, 设计一个算法求两个多项式的和
- 17 给你一个整数数组 `nums` 和一个整数 `k`, 请你返回其中出现频率前 `k` 高的元素。
- 18 +
- 19 两数之和 II - 输入有序数组
- 20 80. 删除有序数组中的重复项 II

## ✧ 矩阵/二维数组

---

- 1 给定一个n行m列的矩阵A (二维数组), 将其水平翻转
- 2 题述同上, 将其垂直翻转
- 3 题述同上, 求A的转置
- 4 题述同上, 旋转90°
- 5 给定一个n行m列的矩阵A和一个m行n列的矩阵B, 求A\*B
- 6 +
- 7 给定一个mxn的矩阵, 找出所在行所在列都为最小值的元素输出其坐标
- 8 给定一个mxn的矩阵, 从外向里以顺时针的顺序依次打印数字 剑指 Offer 29. 顺时针打印矩阵
- 9 36. 有效的数独

## ✧ 单链表 (带头节点/不带头节点)

---

- 1 删除单链表中值为x的前驱节点
- 2 头插法创建一个单链表
- 3 尾插法创建一个单链表
- 4 给定一个单链表，将其逆置（反转）
- 5 将两个递增单链表A和B合并为一个递增单链表C
- 6 设计一个算法使得单链表A递增有序排列
- 7 删除给定单链表中重复值的节点
- 8 给定两个单链表A和B，分别表示两个集合，求他们的交集
- 9 题述如上，求他们的并集
- 10 删除给定单链表的倒数第N个节点
- 11 给定两个带头节点的单链表A和B，若A， B 中全部对应节点的data值相等则返回 1， 否则返回 0
- 12 给定一个单链表A，每 k 个节点一组进行翻转。k 小于等于链表长度。如果节点总数不是 k 的整数，那么最后剩余的节点保持原有顺序。 25. K 个一组翻转链表
- 13 1721. 交换链表中的节点
- 14 24. 两两交换链表中的节点
- 15 剑指 Offer II 022. 链表中环的入口节点
- 16 +
- 17 86. 分隔链表

## ❖ 双链表

---

- 1 头插法创建一个双链表
- 2 给定一个双链表，在值为 x 的节点前面插入元素 y 节点
- 3 给定一个双链表L，找到第一个值为 x 的节点并与后续节点进行交换
- 4

## ❖ 递归

---

- 1 设计一个递归算法求数组A中的最大元素
- 2 求链表中的最大整数（递归）
- 3 求链表的节点个数（递归）
- 4 求链表中所有整数的平均值（递归）
- 5 设计一个递归算法求n!
- 6 判断一个字符串是否为回文串 e.g. abba/abdba

### 7 迭代+递归

(1) 求解二次方根  $\sqrt{A}$  的迭代函数定义如下:

$$\text{sqrt}(A, p, e) = \begin{cases} p & |p^2 - A| < e \\ \text{sqrt}\left(A, \frac{p + \frac{A}{p}}{2}, e\right) & |p^2 - A| \geq e \end{cases}$$

式中,  $p$  是  $A$  的近似二次方根;  $e$  是结果允许误差。试写出相应的递归算法和非递归算法 (假设取绝对值函数  $\text{fabs}()$  可以直接调用)。

### 8 (9) 已知 Ackermann 函数定义如下:

$$\text{Ack}(m, n) = \begin{cases} n + 1 & \text{当 } m = 0 \text{ 时} \\ \text{Ack}(m - 1, 1) & \text{当 } m \neq 0, n = 0 \text{ 时} \\ \text{Ack}(m - 1, \text{Ack}(m, n - 1)) & \text{当 } m \neq 0, n \neq 0 \text{ 时} \end{cases}$$

① 写出计算  $\text{Ack}(m, n)$  的递归算法, 并根据此算法给出  $\text{Ack}(2, 1)$  的计算过程。

- 9 给定一个由不同的小写字母组成的字符串, 输出这个字符串的所有全排列。
- 10 把1~n这n个整数排成一行后随机打乱顺序, 输出所有可能的次序。(全排列)
- 11 从1~n这n个整数中随机选取任意多个, 输出所有可能的选择方案。
- 12 从1~n这n个整数中随机选出m个, 输出所有可能的选择方案。

## \* 栈

- 1 共享栈的实现
- 2 设计一个算法逆向输出单链表所有元素
- 3 设计一个算法将十进制数转化为二进制数
- 4 设计一个算法将十进制数转化为n进制数,  $n < 10$
- 5 括号匹配 检查一个括号序列是否合法, 例如 `[]()[]()` 合法的括号序列满足: 当前括号为左括号时, 下一个是左括号, 或者是对应状态的右括号。

- 6 使用两个队列实现一个栈

## ✧ 队列

---

- 1 循环队列的实现
- 2 使用两个栈实现一个队列

## ✧ 字符串

---

- 1 1768. 交替合并字符串
- 2 将字符串转化为数字 `"123"->123(int), "41231"->41231`
- 3 两个字符串的比较
- 4 设计一个算法判断顺序串s中字母是否唯一
- 5 给你两个二进制字符串 `a` 和 `b`，以二进制字符串的形式返回它们的和。
- 6 151. 反转字符串中的单词
- 7 给定两个字符串 `s` 和 `t`，只包含小写字母。`t` 由 `s` 随机重排，然后在随机位置添加一个字母。请找出在 `t` 中被添加的字母。
- 8 `+`
- 9

## ✧ 二叉树

---

- 1 先，中，后序遍历，递归+非递归
- 2 层次遍历
- 3 设计算法查找二叉树中值为x的节点
- 4 求二叉树所有节点值之和
- 5 求二叉树先序遍历中第k个节点的值
- 6 输出二叉树中的最小值
- 7 求二叉树中叶子节点的个数

- 8 设计算法求二叉树的深度（高度）（自顶向下/自底向上）
- 9 求二叉树中节点值为x的层号
- 10 求出二叉树指定层的叶子节点个数
- 11 找出二叉树中距离根节点最近的叶子节点，若有多个输出任意一个即可（最小深度）（自顶向下/自底向上）
- 12 输出二叉树中最深的元素
- 13 给定两颗二叉树的根节点p和q，判断这两棵树是否相同
- 14 617. 合并二叉树
- 15 寻找二叉树中节点值为x的双亲节点
- 16 判断一颗二叉树是否轴对称
- 17 给定一颗二叉树，判断是否存在从根节点到叶子节点的路径，在此路径上所有节点的值相加等于x
- 18 114. 二叉树展开为链表（难）
- 19 129. 求根节点到叶节点数字之和
- 20 求二叉树的宽度
- 21 给定一颗二叉树，输出每一层节点的平均值
- 22 199. 二叉树的右视图
- 23 103. 二叉树的锯齿形层序遍历
- 24 236. 二叉树的最近公共祖先（难）
- 25 给定一颗二叉树，输出节点值为x的所有祖先节点
- 26 给定一颗二叉树，分行输出所有根节点到叶子节点的路径
- 27 给定一颗二叉树，分行输出所有叶子节点到根节点的路径
- 28 求给定二叉树最长路径长度（即二叉树直径）
- 29 求二叉树中从根节点到叶子节点和为sum的路径，输出此路径（从上到下）
- 30 993. 二叉树的堂兄弟节点
- 31 判断x和y节点是否为兄弟节点
- 32 判断一颗二叉树是否为满二叉树
- 33 判断一颗二叉树是否为完全二叉树
- 34 从下到上，从右到左遍历二叉树
- 35 235. 二叉搜索树的最近公共祖先

36 +

37 [655. 输出二叉树](#)

38 [1123. 最深叶节点的最近公共祖先](#)

39 [105. 从前序与中序遍历序列构造二叉树](#) （重建二叉树）

## ✧ 搜索

---

1 n皇后问题

2 [841. 钥匙和房间](#) （宽搜）

3 [200. 岛屿数量](#)

4 [695. 岛屿的最大面积](#)

5 [542. 01 矩阵](#)

6 [733. 图像渲染](#)

## ✧ 图

---

注意在力扣平台编写时不要使用全局变量，采用局部变量+手动初始化为0

如果没有说明是无向图还是有向图，那就随便

1 求图G的入度和出度（邻接矩阵+邻接表）

2 将邻接矩阵表示方式转换为邻接表（反过来应该也没问题）

3 DFS遍历输出图

4 BFS遍历输出图

5 判断一个无向图是否是一棵树

6 判断一个无向图是否连通/求出连通分量个数/判断节点i是否可以到达所有点

7 判断无向图中节点i和节点j之间是否存在长度为k的简单路径

8 判断无向图中有无环

9 判断有向图是否有环

10 判断无向图节点i和节点j之间是否可达

- 11 997. 找到小镇的法官
- 12 207. 课程表（拓扑排序）
- 13 743. 网络延迟时间（dijkstra）先不写
- 14 1584. 连接所有点的最小费用（kruskal）先不写
- 15 547. 省份数量（并查集）