



数据结构百题狂练

北街学长倾力之作

作者：北街

时间：2025/07/31

版本：1.0



目录

第一章 顺序表	1
第二章 单链表	4
第三章 循环单链表、非循环双链表、循环双链表	13
第四章 栈和队列	16
第五章 树、二叉树	19
第六章 图	27
第七章 查找与排序	32

第一章 顺序表

1. 从顺序表中删除具有最小值的元素（假设唯一），并由函数返回被删除的值，空出的位置有最后一个元素代替，若顺序表为空则显示错误信息并退出运行。

5	3	1	2	7	9
---	---	---	---	---	---

2. 删除顺序表中值为 x 的元素 (x 可以不止一个), 例如 x=2

before delete:

5	3	1	2	7	2
---	---	---	---	---	---

after delete:

5	3	1	7
---	---	---	---

3. 删除有序顺序表中重复的值，使表中所有的元素不同。

before delete:

1	2	2	3	3	5
---	---	---	---	---	---

after delete:

1	2	3	5
---	---	---	---

4. 从有序顺序表中删除给定区间 s - t 之间的值 (包含 s 和 t)，若给定区间不合理或顺序表为空，则显示错误信息并退出。

before delete:

1	2	2	3	3	5
---	---	---	---	---	---

给定: $s=1$, $t=3$

after delete:

5

5. 假设表示集合的顺序表是一个有序顺序表，设计一个高效的算法实现集合的求差集运算，即 $C=A-B$

A:

1	2	2	3	3	4
---	---	---	---	---	---

B:

2	2	2	3	3	5
---	---	---	---	---	---

C:

1	4
---	---

6. 【2011 统考真题】一个长度为 $L(L \geq 1)$ 的升序序列 S ，处在第 $L/2$ 个位置的数称为 S 的中位数。例如，若序列 $S_1=(11,13,15,17,19)$ ，则 S_1 的中位数是 15，两个序列的中位数是含它们所有元素的升序序列的中位数。例如，若 $S_2=(2,4,6,8,20)$ ，则 S_1 和 S_2 的中位数是 11。现在有两个等长升序序列 A 和 B ，试设计一个在时间和空间两方面都尽可能高效的算法，找出两个序列 A 和 B 的中位数。

A:

1	2	4	6	7	8
---	---	---	---	---	---

B:

4	5	7	9	11	13
---	---	---	---	----	----

7. 将两个升序顺序表合并成一个新的顺序表，并由函数返回新的结果顺序表。

A:

1	2	4	6	7	8
---	---	---	---	---	---

B:

4	5	7	9	11	13
---	---	---	---	----	----

after merge:

1	2	4	5	6	7	8	9	11	13
---	---	---	---	---	---	---	---	----	----

8. 实现一个空间复杂度为 $O(1)$ 的顺序表逆置算法。

before reverse:

1	2	4	6	7	8
---	---	---	---	---	---

after reverse:

8	7	6	4	2	1
---	---	---	---	---	---

9. 将 arr 中的存放的数据循环左移 p 个位置,即将 arr 中的 $(X_0, X_1, \dots, X_{n-1})$ 变换为 $(X_p, X_{p-1}, \dots, X_{n-1}, X_0, X_1, \dots, X_{p-1})$), 尽可能高效

p=3:

1	2	4	6	7	8
---	---	---	---	---	---

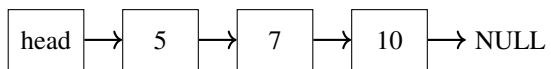
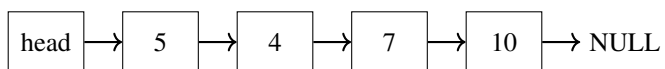
p=3:

6	7	8	1	2	4
---	---	---	---	---	---

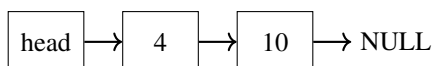
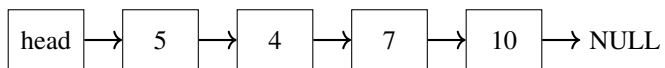
第二章 单链表

1. 单链表的基本操作：创建一个空链表、创建有值的单链表、输出单链表、输出单链表的长度、判断单链表是否为空、输出单链表的第 k 个元素、输出指定元素的位置、在第 k 个元素位置上插入 f 元素、删除单链表的第 k 个元素、释放单链表。

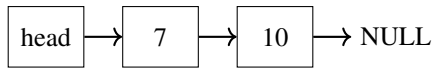
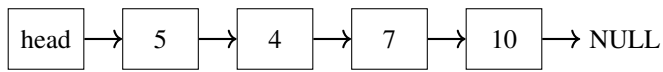
2. 删除单链表中的唯一的最小值。



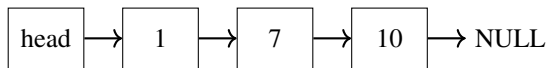
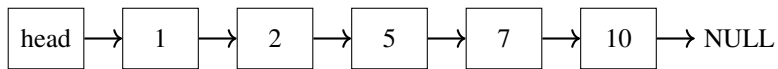
3. 删除单链表中序号为奇数的结点（序号从首元结点开始）。



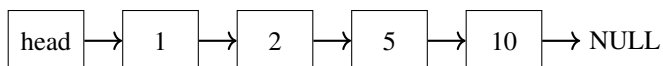
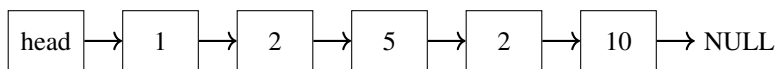
4. 设一个带头结点的单链表所有元素的数据值**无序**，试编写函数删除表中介于给定的两个值（作为函数参数给出）之间的元素。例如，删除表中介于 2 和 7 之间的元素，如下图所示：



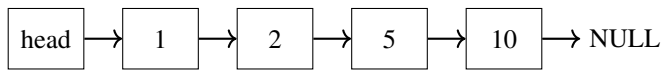
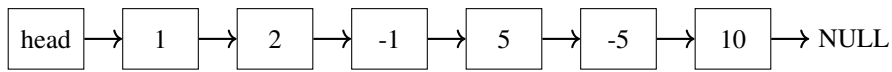
5. 设一个带头结点的单链表所有元素的数据值**递增有序**，试编写函数删除表中介于给定的两个值（作为函数参数给出）之间的元素。例如，删除表中介于 2 和 7 之间的元素，如下图所示：



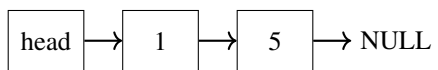
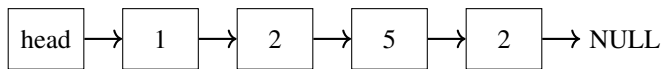
6. 在一个**递增有序**的线性表中，有数值相同的元素存在，若存储方式为单链表，设计算法去掉数值相同的元素，使表中不再有重复的元素。



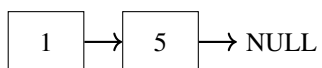
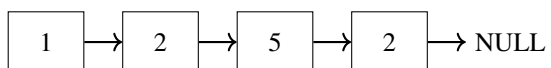
7. 用单链表保存 m 个整数，结点的结构为 $[\text{data}][\text{next}]$ ，且 $|\text{data}| \leq n$ 。现要求设计一个时间复杂度尽可能高效的算法，对于链表中绝对值相等的结点，仅保留第一次出现的结点而删除其余绝对值相等的结点。



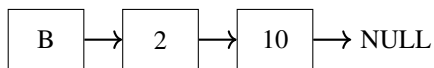
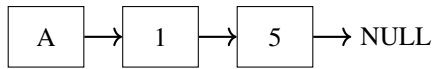
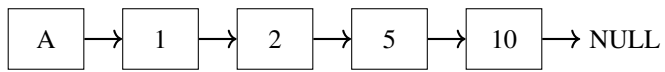
8. 在带头结点的单链表 L 中，删除所有值为 X 的结点，并释放其空间，假设值为 X 的结点不唯一



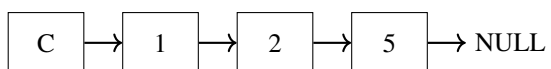
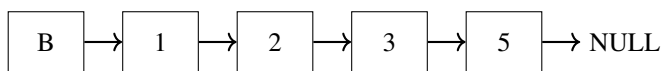
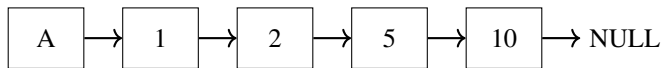
9. 设计一个递归算法，删除一个不带头结点的单链表中所有值为 x 的结点



10. 将一个带头结点单链表 A 分解成两个带头结点的单链表 A 和 B，使得 A 中含有原表中序号为奇数的元素，B 中含有原表中序号为偶数的元素，且保持其相对位置不变。

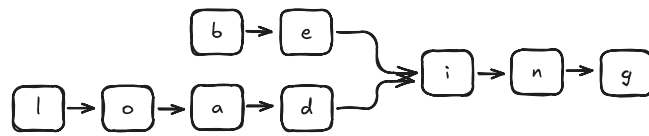


11. 设 A、B 是两个单链表（带头结点），其中元素**递增有序**，设计一个算法从 A、B 中的公共元素产生单链表 C，要求不破坏 A、B 结点

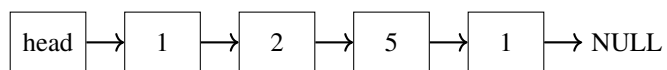


12. 存在这样一种情况，如果两个单词有相同的后缀，那我们可以将后缀作为公共部分存储，比如 being 和 loading，其中 ing 就可以作为公共部分，现在存在两个链表，含有公共部分，设计一个高效算法找到其公

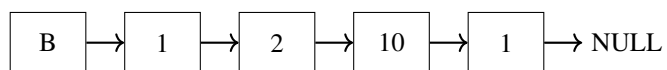
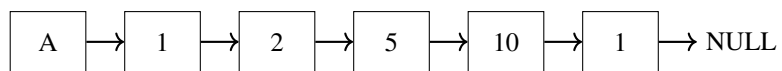
共后缀真实位置。



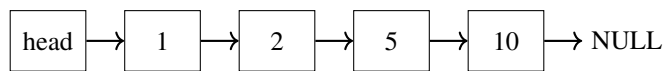
13. 设计一个算法找到单链表中最后一个最小值。例如：下面的链表中，最后一个最小值是 1，尾节点。



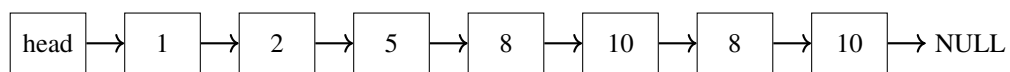
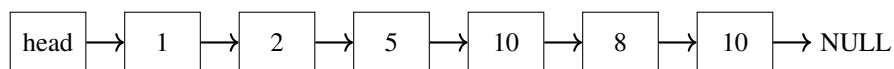
14. 求单链表中的中间位置结点元素，结点为偶数则向上取整。例如：下面的 A 链表中，中间位置结点是 5，B 链表中，中间位置结点是 2。



15. 有一个带头结点的单链表，设计一个函数找到指定的倒数第 k 个结点，输出结点值，并返回 1，否则返回 0，前提不能改变链表，尽可能高效。例如：下面的链表中，倒数第 2 个结点是 5。

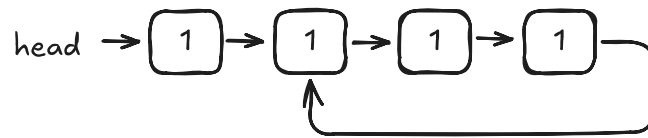


16. 设计算法在单链表中的第一个最大值前面插入值为 x 的结点。例如：下面的链表中，第一个最大值是 10，在 10 前面插入值为 8 的结点。

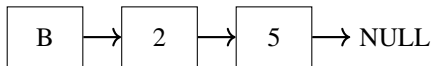
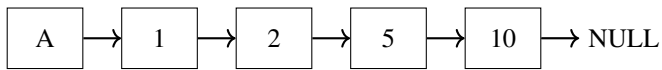


17. 判断单链表是否是升序（带头结点）。

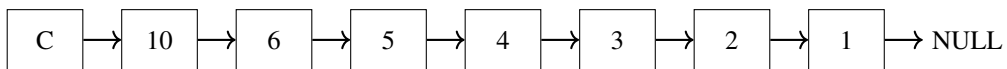
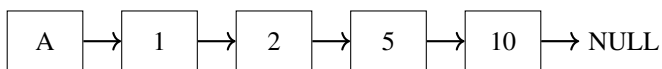
18. 设计一个算法判断一个单链表是否有环。



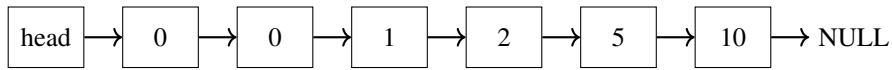
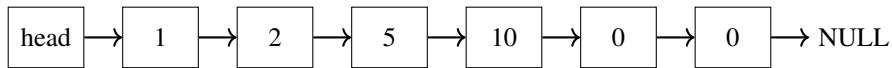
19. 存在两个单链表序列 A、B，设计函数判断 B 是否为 A 的子序列。例如：下面的链表中，B 链表是 A 链表的子序列。



20. 假设有两个按元素值递增次序排列的线性表，均以单链表形式存储。请编写算法将这两个单链表归并为一个按元素递减的单链表，并要求利用原来两个单链表的结点存放归并后的单链表。如下面的链表 A 和链表 B，归并后的链表 C 为 10,6,5,4,3,2,1。注意链表 C 并不是新创建的，而是利用了原来两个链表的结点。

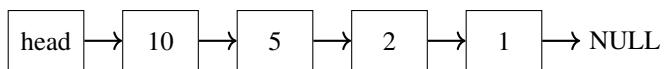
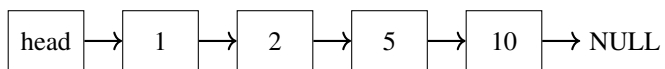


21. 将单链表里的所有零元素移动到非零元素前面。例如：下面的链表中，将所有零元素移动到非零元素前面。



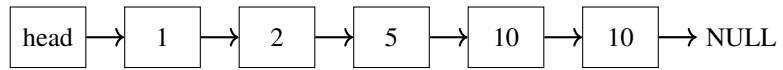
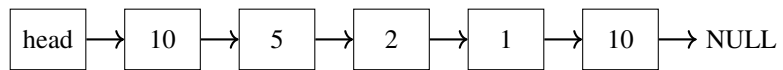
22. 按递增次序输出单链表各结点的数据元素，并释放结点所占的存储空间。

23. 设计一个算法就地逆置单链表。所谓就地指的是空间复杂度为 $O(1)$ 。



24. 有一带头结点的单链表，设计一算法从尾到头的输出每个结点的值。

25. 有一个带头结点的单链表 L，设计一个算法使其递增有序。注意：不能仅仅交换数据域，而是要交换结点。



第三章 循环单链表、非循环双链表、循环双链表

1. 创建循环单链表。
2. 设有一个带头结点的循环单链表，其结点均为正值，设计一个算法，反复找出单链表中最小值输出并删除后释放，直到单链表为空，最后释放头结点。
3. 已知带头结点的循环单链表 L 中至少有两个结点，每个结点的两个域为 `data` 和 `next`，其中 `data` 的类型为整型。设计算法判断该链表中的每个值是否小于其后续两个结点之和。
4. 有两个循环单链表，链表头指针分别为 `h1` 和 `h2`，编写一个函数将链表 `h2` 连接到 `h1` 之后，要求连接后的链表仍保持循环链表形式。

5. 创建双向链表。

6. 设计算法在带头结点循环双链表删除 data 域为 x 的结点。

7. 有一个非空双向链表 L，设计算法在第 k 个结点之后插入一个值为 x 的结点。

8. 设头指针为 L 的带有表头结点的非循环双向链表，其每个结点中除有 pred(前驱指针)、data(数据)和 next(后继指针)域外，还有一个访问频度域 freq。在链表被启用前，其值均初始化为零。每当在链表中进行一次 Locate(L,x) 运算时，令元素值为 x 的结点中 freq 域的值增 1，并使此链表中结点保持按访问频度非增(递减)的顺序排列，同时最近访问的结点排在频度相同的结点前面，以便使频繁访问的结点总是靠近表头。试编写符合上述要求的 Locate(L,x) 运算的算法，该运算为函数过程，返回找到结点的地址，类型为指针型。

9. 非循环带头结点双链表中，将 p 所指结点与其后继结点交换（ p 非尾结点）。

10. 设计一个算法判断带头结点的循环双链表是否对称。

11. 设计算法就地逆置双链表。就地是指空间复杂度为 $O(1)$ 。

12. 带头结点双链表，设计算法使其元素递增有序。

第四章 栈和队列

1. 顺序栈、链栈的创建及其基本操作。

2. 链队、循环队列的创建及其基本操作。

3. 利用一个栈实现以下递归函数的非递归计算。

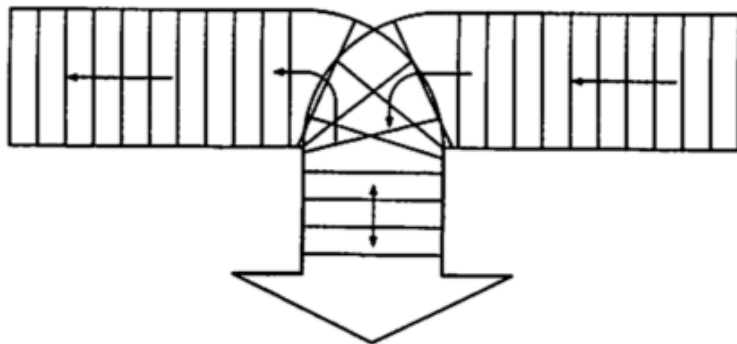
$$P_n(x) = \begin{cases} 1, & n = 0 \\ 2x, & n = 1 \\ 2xP_{n-1}(x) - 2(n-1)P_{n-2}(x), & n > 1 \end{cases}$$

4. 设单链表的表头指针为 h ，结点结构由 $data$ 和 $next$ 两个域构成，其中 $data$ 域为字符型。试设计算法判断该链表的全部 n 个字符是否是中心对称，例如 xyx ， $xyyx$ 都是中心对称。

-
5. 某汽车轮渡口，过江渡船每次能载 10 辆车过江。过江车辆分为客车类和货车类，上渡船有如下规定：同类车先到先上船；客车先于货车上船，且每上 4 辆客车，才允许放上 1 辆货车；若等待客车不足 4 辆，则以货车代替；若无货车等待，允许客车都上船。试设计一个算法模拟渡口管理。
 6. 假设一个算法表达式包含圆括号、方括号、和花括号 3 种类型的括号，编写一个算法来判断表达式里的括号是否配对，以字符 '\0' 作为算术表达式的结束符。
 7. 有一个队列和一个栈，设计一个算法使队列中的元素逆置。
 8. 设有两个栈 s1、s2 都采用顺序栈方式，并共享一个存储区，大小为 max，为了尽量利用空间，减少溢出的可能，可采用栈顶相向、迎面增长的存储方式，试设计 s1、s2 有关入栈和出栈的算法。

9. 若希望循环队列中的元素都能得到利用，则需设置一个标志域 **tag**，并以 **tag** 的值为 0 或 1 来区分队头指针 **front** 和队尾指针 **rear** 相同时的队列状态是“空”还是“满”。试编写此结构相应的入队和出队算法。

10. 按下图所示铁道进行车厢调度（注意，两侧铁道均为单向行驶道，火车调度站有一个用于调度的“栈道”），火车调度站的入口处有 n 节硬座和软座车厢（分别用 **H** 和 **S** 表示）等待调度，试编写算法，输出对这 n 节车厢进行调度的操作（即入栈或出栈操作）序列，以使所有的软座车厢都被调整到硬座车厢之前。



第五章 树、二叉树

1. 已知一棵二叉树按顺序存储结构进行存储，设计一个算法，求编号分别为 i 和 j 的两个结点的最近的公共祖先结点的值。
2. 二叉树的创建及其基本操作：先序递归、中序递归、后序递归。
3. 利用递归遍历的简单应用：统计结点个数、输出所有叶子结点、求结点值为 x 的所在层次、求第 k 层结点个数、求指定值为 x 的结点的双亲结点 p 、求二叉树中的最大结点值。
4. 试写出先序遍历的非递归算法。

5. 试写出中序遍历的非递归算法。

6. 试写出后序遍历的非递归算法。

7. 试写出层次遍历的算法。

8. 假设二叉树是用二叉链表存储，试设计一个算法, 求先序遍历中第 k ($1 \leq k \leq$ 二叉树的结点个数) 个结点的值。

9. 求二叉树中距离根结点最近的叶子结点。

10. 假设二叉树采取二叉链表存储结构存储，试设计一个算法，计算一颗给定的二叉树所有的双分支结点个数。

11. 设计一个算法复制一颗二叉树。

12. 设计一个算法将二叉树的叶结点按从左到右的顺序连成一个单链表，表头指针为 `head`。二叉树按二叉链表方式存储，连接时用叶结点的右指针来存放单链表指针。

13. 设一个二叉树各结点的值互不相同，其先序遍历序列和中序遍历序列分别存于两个一维数组 `A`、`B` 中，试编写算法建立该二叉树的二叉链表。

14. 一颗二叉树以二叉链表的形式存储，编写一个算法判断其是否是一个完全二叉树。

15. 采用递归的方式求二叉树的高度。

16. 假设二叉树采用二叉链表存储结构，设计一个非递归算法求二叉树的高度。

17. 已知二叉树以二叉链表存储，编写算法完成：对于树中每一个元素值为 x 的结点，删除以它为根的子树，并释放相应的空间。

18. 请设计一个算法，将给定的表达式树，转换成等价的中缀表达式并输出。

19. 假设二叉树采用二叉链表存储结构，设计一个算法，求非空二叉树的宽度（即具有结点数最多的那一层的结点数）。

-
20. 假设二叉树采取二叉链表存储结构存储，试编写算法打印值为 x 的结点的所有祖先，假设 x 的值不多于一个。
21. 设有一颗满二叉树（所有结点值均不同），已知其先序序列为 pre ，设计一个算法求其后序序列 $post$ 。
22. p 、 q 分别为指向该二叉树中任意两个结点的指针，试编写算法 $ancestor(root, p, q, r)$ ，找到 p 、 q 的最近公共祖先结点 r 。
23. 编写算法打印从根结点到所有叶子结点的路径。
24. 设计算法，输出二叉树中第一条最长的路径长度，并输出此路径上各结点的值。

-
25. 试编写一个算法将一颗二叉树的所有结点的左右子树进行交换。
26. 二叉树的带权路径长度 (WPL) 是二叉树中所有叶结点的带权路径长度之和。给定一颗二叉树 T , 采用二叉链表存储, 结点结构为 `left weight right` 试设计求 T 的 WPL 的算法。
27. 编写一个算法判断给定的二叉树是否是二叉排序树。
28. 设计一个算法, 求出指定结点在给定二叉排序树中的层次。
29. 利用二叉树遍历的思想编写一个判断二叉树是否是二叉平衡树的算法。

-
30. 编写一个递归算法，在一棵有 n 个结点的、随机建立起来的二叉排序树上查找第 k 小的元素，并返回指向该结点的指针，要求算法的平均时间复杂度为 $O(\log n)$ 。二叉排序树的每个结点中除了 `data`、`lchild`、`rchild` 等数据成员外，增加一个 `count` 成员，保存以该结点为根的子树的结点个数。
31. 编程求以孩子兄弟表示法存储的森林的叶子结点数。
32. 以孩子兄弟链表为存储结构，请设计递归算法求树的高度。
33. 从所给的树的层次遍历及其结点度数信息，构造孩子兄弟链表。
34. 将二叉树先序线索化、中序线索化、后序线索化。

35. 写出在中序线索二叉树里查找指定结点在后序的前驱结点的算法。

36. 哈夫曼树及哈夫曼编码实现。

第六章 图

1. 以邻接矩阵方式创建一幅图。
2. 以邻接表方式创建一幅图。
3. 写出从图的邻接表表示转换成邻接矩阵表示的算法。
4. 广度优先遍历（邻接矩阵与邻接表）。
5. 深度优先遍历（邻接矩阵与邻接表）。

6. 写出深度优先遍历的非递归算法（图用邻接表给出）。

7. 最小生成树之 prime 算法。

8. 最小生成树之 kruskal 算法。

9. 利用广度优先遍历求解无权图单源最短路径。

10. 最短路径之 dijkstra（邻接矩阵与邻接表）。

11. 最短路径之 floyd。

12. 采用深度优先遍历判断是否存在由 v_i 到 v_j 的路径，图用邻接表存储。

13. 采用广度优先遍历判断是否存在由 v_i 到 v_j 的路径，图用邻接表存储。

14. 输出顶点 v_i 到顶点 v_j 的所有简单路径，图用邻接表存储。

15. 输出顶点 v_i 到顶点 v_j 的长度为 L 的所有简单路径，图用邻接表存储。

16. 输出经过顶点 k 的所有简单回路，图用邻接表存储。

17. 【2021 统考真题】已知无向连通图 G 由顶点集 V 和边集 E 组成， $|E| > 0$ ，当 G 中度为奇数的顶点个数为不大于 2 的偶数时， G 存在包含所有边且长度为 $|E|$ 的路径（称为 EL 路径）。设图 G 采用邻接矩阵存储，设计算法，判断是否存在 EL 路径。

18. 在采用邻接表存储的图上统计连通分支数并输出每个连通分支的顶点。

19. 邻接矩阵存储的图中设计算法求树的直径。（在一棵树中，每一条边都有权值，树中的两个点之间的距离，定义为连接两点的路径上边权之和，那么树上最远的两个点，他们之间的距离，就被称之为树的直径。）

20. 邻接矩阵存储一个有向图，设计算法求该有向图的所有根。

21. 设计一个算法，判断一个无向图是否是一棵树。

22. 设计一个算法实现拓扑排序（图以邻接表方式给出）

第七章 查找与排序

1. 折半查找（递归与非递归）。
2. 直接插入算法实现。
3. 希尔排序算法实现。
4. 简单选择排序算法实现。
5. 简单选择排序在单链表中使用。

6. 冒泡排序算法实现。

7. 双向冒泡算法实现。

8. 快速排序算法实现。

9. 有一数组存放着无序序列 $K_1, K_2, K_3, \dots, K_n$ ，现要求将 K_n 放在将元素排序后的正确位置上，试编写实现该功能的算法，要求比较关键字的次数不超过 n 。

10. 堆排序算法实现。

11. 编写算法判断一个序列是否是小根堆。

12. 归并排序算法实现。

13. 计数排序算法实现。

14. 编写一个算法，使之能够在数组中找到第 k 小的元素。

15. 设计一个算法：将顺序表中的奇数全部移到偶数前面（时间最少，空间最少）。

16. 2016 年 408 真题：集合 A 划分为 A_1 、 A_2 ，个数分别为 n_1 、 n_2 ，和分别为 S_1 、 S_2 ， $|n_1 - n_2|$ 最小 $|S_1 - S_2|$ 最大。