

线性数据结构部分：

一、填空题

1. 数据结构包括数据的逻辑结构、 数据的储存结构 和数据的运算三个方面。
2. 数据结构包括 逻辑结构 、数据的存储结构和数据的运算三个方面。
3. 在算法“正确”的前提下，评价算法主要有两个指标是：时间复杂度和 空间复杂度 。
4. 在算法“正确”的前提下，衡量算法效率的主要指标是： 时间复杂度 及空间复杂度。
5. 线性数据结构的逻辑特征是有且仅有一个 开始结点 和一个终端结点，且所有结点都最多只有一个直接前趋和一个 直接后继 。
6. 线性数据结构的逻辑特征是有且仅有一个开始结点和一个终端结点，且所有结点都最多只有一个直接前趋 和一个直接后继。
7. 数据的存储结构包含有 顺序存储结构 、 链式存储结构 、 索引存储结构 和 散列存储结构 等四种基本的映像方法。
8. 数据结构的四种基本形式是： 顺序 存储结构、 链式 存储结构、索引存储结构以及散列存储结构。
9. 若频繁地对线性表进行插入与删除操作，该线性表应采用 链式 存储结构。
10. 线性链表中数据元素的组成：一是数据元素的值，二是直接后继元素的 地址的指针 ，这两部分信息组成数据元素的存储映像，即结点。
11. 带头结点的单向链表L为空的判定条件是 $L \rightarrow next == null$ 。
12. 在一个单链表中p所指结点之后插入s所指结点时，应执行 $s \rightarrow next = p \rightarrow next$ 和 $p \rightarrow next = s$ 的操作。
13. 要在一个单链表中p所指结点之后插入一个子链表，子链表第一个结点的地址为s，子链表最后一个结点的地址为t，则应执行操作：

$t \rightarrow next = p \rightarrow next$ 和 $p \rightarrow next = s$ 。

14. 单链表的结点的数据类型是：

```
typedef struct node{
    int data;    /*数据部分*/
    struct node *next; /*指向下一个结点的指针*/
} LinkList;
LinkList *p, *q;
```

如果要求将由指针变量q所指向的表外结点插入到单链表中由p所指向的结点之后，则应执行的语句

是：(1) $q \rightarrow next = p \rightarrow next$ (2) $p \rightarrow next = q$ 。要将p所指向的结点的数据部分修改为25，应执行的语句是： $p \rightarrow data = 25$ 。

15. 插入和删除只允许在表的同一端进行的线性表称为 栈 ，它具有 后进先出 的特性。

16. 将插入操作限定在表的一端而删除操作限定在表的另一端的线性表称为 队列 ，它具有 先进先出 的特性。

17. 对于一个以顺序存储实现的循环队列 $Q[0..10]$ ，队头、队尾的位置指示器分别是front, rear, 初始时都被设置为-1，则在该循环队列中实现出队操作时，判空的条件是： $front == rear$ ；入队操作时，判满的条件是： $front == (rear + 1) \% 11$ 。

18. 二维数组 $A[10][20]$ 采用列序为主方式存储，每个元素占10个存储单元，且 $A[0][0]$ 的存储地址是2000，则 $A[6][12]$ 的地址是 3320 。

19. 已知二维数组 $A[20][10]$ 采用行序为主方式存储，每个元素占2个存储单元，并且 $A[10][5]$ 的存储地址是1000，则 $A[18][9]$ 的存储地址是 1168 。

20. 线性表的三种基本查找方法是：顺序查找、 折半 查找和 分块 查找。

二、单项选择题

1. 线性表中 (D) 称为线性表的长度。

A、元素的长度 B、数据项的数目

C、数据的长度 D、元素的个数

2. 不属于线性表基本运算的是： (B)。

A、删除运算 B、指针运算 C、取结点运算 D、插入运算

3. 在下列关于线性表的叙述中，错误的是： (B)。

A、采用顺序存储的线性表，必须占用一片连续的存储单元

B、采用顺序存储的线性表，便于进行插入和删除操作

C、采用链式存储的线性表，不必占用一片连续的存储单元

D、采用链式存储的线性表，便于进行插入和删除操作

4. 当线性表选择链表作为存储结构时，不具有的特点是： (A)。

A、可随机访问任一元素 B、插入、删除时不需要移动大量元素

C、不必事先估计存储空间 D、所需空间与线性表的长度成正比

5. 算法具有“确定性”等5个特性，下面对另外4个特性的描述中错误的是 (D)。

A、可行性 B、有零个或多个输入 C、有穷性 D、有零个或多个输出

6. 衡量一个算法的质量除了正确性之外，最重要的是要考查 (C)。

A、可行性 B、有穷性 C、时间复杂度和空间复杂度 D、输入和输出

7. 在长度为n的线性表中, 在第i个元素之前插入一个新的元素x, 需要移动 (B) 个元素。

A、n B、n-i+1 C、n-i D、i+1

8. 假设p是指向线性表中第i个数据元素结点的指针, 则p->next是指向第i+1个数据元素结点的指针, 若p->data=ai, 则p->next->data=ai+1, 那么p->next->next指向的是第 (C) 个结点。

A、i B、i+1 C、i+2 D、i+3

9. 以下哪一个不是队列的基本运算? B

A、从队尾插入一个新元素 B、从队列中删除第i个元素

C、判断一个队列是否为空 D、读取队头元素的值

10. 在初始为空的队列中顺序插入元素a,b,c,d以后, 紧接着作了两次删除操作, 此时的队尾元素是 (D)。

A、a B、b C、c D、d

11. 队列的顺序存储方式中判断循环队列为满的条件是 (C)。

A、front==rear B、front==(rear+1)%(maxsize+1)

C、front==(rear+1)%maxsize D、front==rear%maxsize

12. 单链表Head中, 在指针q所指结点后面插入一个由指针P所指结点, 则执行(D)。

A、q->next=p->next; p->next=q; B、p->next=q->next; q=p;

C、q->next=p->next; p->next=q; D、p->next=q->next; q->next=p;

13. 一个栈的输入序列是1, 2, 3, 4, 则下列序列中不可能是栈的输出序列的是(D)。

A、1234 B、4321 C、2341 D、4123

14. 设在栈中, 由顶向下已存放元素c, b, a, 在第四个元素d入栈前, 栈中元素可以出栈。试问在d入栈后, 不可能的出栈序列是: (C)。

A、d c b a B、c b d a C、c a d b D、c d b a

15. 栈S最多能容纳4个元素。现有6个元素按A、B、C、D、E、F的顺序进栈, 问下列哪一个序列是可能的出栈序列? (C)

A、E D C B A F B、B C E F A D C、C B E D A F D、A D F E B C

16. 设一个栈的入栈序列是abcde, 则在下列输出序列中不可能的出栈序列是: (C)

A、e d c b a B、d e c b a C、d c e a b D、a b c d e

17. 若用一个大小为6的数组来实现循环队列, 且当前rear和front的值分别为0和3。当从队列中删除一个元素, 再加入两个元素后, rear和front的值分别为 (B)。

A、1和5 B、2和4 C、4和2 D、5和1

18. 设有9个数据记录组成的线性表，它们的排序键码字的取值分别是（11，15，20，27，30，35，46，88，120），已经将它们按照排序码递增有序的方式存放在一维结构数组a[0..8]中从下标0开始到下标8结束的位置，则当采用折半查找算法查找关键字值等于20的数据记录时，所需比较的元素的下标依次是：（ B ）。（注：计算中间位置时取下整）

A、0，1，2 B、4，1，2 C、4，2 D、4，3，2

19. 采用折半查找方法进行查找的数据文件应满足的条件是：（ C ）。

A、顺序存储 B、链式存储 C、顺序存储且已排序 D、链式存储且已排序

三、判断题

1. （ B ）单链表从任何一个结点出发，都能访问到所有结点。
2. （ A ）顺序表是一种随机存取的存储结构。
3. （ B ）线性表的逻辑顺序与存储顺序总是一致的。
4. （ B ）线性表的链式存储结构优于顺序存储结构。
5. （ A ）数据的存储结构是数据的逻辑结构在存储单元中的表示形式。
6. （ B ）程序的执行效率与数据存储结构的选择没有直接的关系。
7. （ B ）线性表的长度是指线性表所占存储空间的大小。
8. （ B ）线性表的长度决定了线性表所占存储空间的大小，但它不等于线性表所占存储空间的大小。
9. （ B ）在采用链式存储结构的线性表上查找某个元素的平均效率比在采用顺序存储结构的线性表上查找的平均效率高
10. （ B ）链式存储结构的线性表适用于对数据进行频繁的查找操作，而顺序存储结构的线性表则适宜于进行频繁地插入、删除操作。
11. （ B ）在单链表中，给定任一结点的地址p，则可用下述语句将新结点s插入结点p的后面：p->next = s；s->next = p->next；
12. （ B ）二维数组是其数据元素为线性表的线性表。
13. （ A ）N（N>1）维数组可以看作是线性表的推广。
14. （ B ）循环队列也存在空间溢出问题。
15. （ B ）队列和栈都是运算受限的线性表，插入或者删除运算只允许在表的同一端进行。
16. （ B ）从数据元素插入、删除的规则来看，队列的本质特征是LIFO，栈的本质特征是FIFO。
17. （ B ）所有插入排序算法均是稳定的。

18. (B) 顺序存储方式只能用于存储线性结构。数组
19. (B) 程序的执行效率只决定于算法设计的技巧,与程序设计中所采用的数据的表示方式及数据逻辑模型的实际存储形式无关。
20. (B) 线性表的特点是每个元素都有一个前驱结点和一个后继结点。
21. (B) 链表的每个结点中都包含一个指针。
22. (B) 算法一定要有输入和输出。
23. (B) 顺序查找算法可以用在顺序存储结构表示的线性表上查找数据元素,但不可以用在链式存储结构的线性表上查找数据元素。
24. (A) 折半查找方法只能用在采用顺序存储结构的有序线性表中来实现对某一数据项的快速查找。
25. (B) 折半查找方法可以用在采用单向链表形式存储的有序线性表中实现对某一数据项的快速查找。
26. (A) 判断某个排序方法的稳定性可以通过一次或几次输入数据序列,看排序结果是否改变了原始待排数据序列中关键字值相同的数据的相对次序,从而作出该排序算法是否稳定的结论。

四、简答题

1. 什么是算法? 具有哪些特性? 如何衡量一个算法的好坏? 算法与程序有何不同?

答: 算法是对特定问题求解步骤的一种描述, 它是指令的有限序列, 其中每一条指令表示一个或多个操作。五个特性: 有穷性 确定性 可行性 输入 输出。评价算法两个指标: 时间复杂度 空间复杂度。算法与程序: (1). 一个程序不一定满足有穷性。例操作系统, 只要整个系统不遭破坏, 它将永远不会停止, 即使没有作业需要处理, 它仍处于动态等待中。因此, 操作系统不是一个算法。(2). 程序中的指令必须是机器可执行的, 而算法中的指令则无此限制。(3). 算法代表了对问题的解, 而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述, 则它就是一个程序。

2. 线性表顺序存储结构的优缺点是什么?

答: 顺序存储结构是在内存中开辟一个连续的空间用来存储数据, 因此对于内存的需求和苛刻, 必须是连续的空间。在数据查找(特别是不按照规律排列的数据), 时间复杂度较少。效率高。

3. 线性结构与非线性结构有何差别?

答: 1. 线性储存主要是时间复杂度比较小, 而且代码可读性也比较好, 非线性结构实现起来很多时候都是以递归实现, 所以能用线性储存的数据最好不用非线性结构。

2. 非线性结构就是存储联系复杂的数据, 这些数据之间一般都有较大的相关性, 线性链表是不能实现的。

4. 简述顺序表和链表之间的差异。

链式存储结构：

- (1) 占用额外的空间以存储指针 (浪费空间)
- (2) 存取某个元素速度慢
- (3) 插入元素和删除元素速度快
- (4) 没有空间限制, 存储元素的个数无上限, 基本只与内存空间大小有关。

顺序存储结构：

- (1) 空间利用率高
- (2) 存取某个元素速度快
- (3) 插入元素和删除元素存在元素移动, 速度慢, 耗时
- (4) 有空间限制, 当需要存取的元素个数可能多于顺序表的元素个数时, 会出现"溢出"问题。当元素个数远少于预先分配的空间时, 空间浪费巨大。

在存取元素频繁, 但删除或插入操作较少的情况宜用顺序表。堆排序, 二分查找适宜用顺序表。

5. 什么是排序方法的稳定性?

排序的稳定性, 就是指, 在对a关键字排序后会不会改变其他关键字的顺序。

比如排序 (2, 3, 1 (第一个), 1 (第二个), 5, 6)

不稳定的排序, 可能会排出

(1 (第二个), 1 (第一个), 2, 3, 5, 6);

而稳定的排序则不会, 在比较的关键字相同的情况下, 稳定的排序会将较早出现的元素排在前面。

6. 什么叫栈? 它有哪些基本操作? 各个基本操作的含义是什么?

栈是限定只能在表的同一端进行插入和删除操作的线性表。

基本操作: `Inistack(S)`: 初始化栈S为空栈

`Empty(S)`: 判定S是否为空栈。若空, 返回真, 非空返回假

`Push(S, x)` 进栈操作。在栈S栈顶插入数据元素x

`Pop(S)` 出栈操作。若栈S不是空栈, 则删除栈顶元素

`Gettop(S)` 取栈顶元素。只读取栈顶元素, 不改变栈中内容。

7. 什么叫队列? 它有哪些基本操作? 各个基本操作的含义是什么?

队列是一种先进先出的线性表。它只允许在表的一段进行插入元素操作, 在另一段进行删除元素操作。

基本操作: `Iniqueue(Queue)`初始化队列Queue, 置Queue为空队列

`Empty(Queue)`判定队列Queue是否为空, 若Queue是空返回真, 若非空返回假

`Addqueue(Queue, x)`入队操作, 若队列为满, 则在Q的队尾插入元素x

`Delqueue(Queue)`出队操作, 若队列非空, 则将Queue的队头元素删除

`Getheadqueue(Queue)`读取队列Queue的队头元素, 不改变队列中的内容。

五、综合题

1.

给出下列稀疏矩阵A的三元组表示法的存储模型。

```
0 1 33
0 3 22
0 5 -15
1 2 3
2 3 -12
3 4 -10
5 5 2
```

2. 给出下列稀疏矩阵所对应的三元组表示方法存储模型。

```
0 1 12
0 2 9
2 0 -3
```

3 5 14
3 2 24
4 1 18
5 0 15
5 3 -7

3. 给出下列稀疏矩阵A的三元组表示法的存储模型。

0 0 12
0 2 10
0 5 9
11 2 3
2 3 25
3 4 -9
5 5 23

4. 设单链表的结点为: `typedef struct node`

`{ int data;`

`struct node *next } LinkList;`

`LinkList *p,q;`

如果要求将由指针变量q所指向的结点插入到单向链接表中p所指向的结点之后, 则应执行的语句是什么? 要将p所指向的结点的数据部分修改为23, 应执行的语句是什么?

`q->next=p->next;p->next=q;p->data=23;`

5. 对于给定的一组关键字:503,087,512,067,908,170,889,276,675,453; 请按关键字递减排序, 写出直接插入排序、冒泡排序的各趟运行结果。

书上的排序方法都是按递增顺序讲的, 我猜递减顺序写法差不多, 就按照葫芦画瓢了

直接插入排序503 087 512 067 908 170 889 276 675 453

这列是监视哨，老师说考试 时候不写要扣分.....还 得花箭头，电脑不会画，看 书上的例子吧

```
[503]087 512 067 908 170 889 276 675 453 087
[503 087]512 067 908 170 889 276 675 453 512
[512 503 087]067 908 170 889 276 675 453 067
[512 503 087 067]908 170 889 276 675 453 908
[908 512 503 087 067]170 889 276 675 453 170
[908 512 503 170 087 067]889 276 675 453 889
[908 889 512 503 170 087 067]276 675 453 276
[908 889 512 503 276 170 084 067]675 453 675
[908 889 675 512 503 276 170 084 067]453 453
```

冒泡排序

```
503  503  512 512    908 908   908 908 908 908
087  512  503 908    512 889   889 889 889 889
512  087  908 503    889 513   675 675 675 675
067  908  176 889    503 675   513 513 513 513
908  173  889 276    675 503   503 503 503 503
170  889  276 675    453 453   453 453 453 453
889  276  675 453    276 276   276 276 276 276
276  675  453 176    176 176   176 176 176 176
675  453  087 087    087   087   087 087 087 087
453  067  067 067    067 067   067 067 067 067
```

6. 某长度为10的有序表存储于一维结构数组中下标0~9的位置中,其中记录的关键键码值依次是:
5,10,18,21,33,47,48,55,80,125,现要查找关键码值为18及83的记录,现规定在中间位置计算时采用“向下取整”的方法,写出折半查找的过程及查找结果。

不知道规范过程怎么写,非规范版大致如下

找18

low=0 high=9

mid=(low+high)/2=4----33 18<33 high=mid-1=3

$\text{mid}=(\text{low}+\text{high})/2=1\text{-----}10 \quad 18>10 \quad \text{low}=\text{mid}+1=2$

$\text{mid}=(\text{low}+\text{high})/2=2\text{-----}18$

查找成功

找83

$\text{low}=0 \quad \text{high}=9$

$\text{mid}=(\text{low}+\text{high})/2=4\text{-----}33 \quad 83>33 \quad \text{low}=\text{mid}+1=5$

$\text{mid}=(\text{low}+\text{high})/2=7\text{-----}55 \quad 83>55 \quad \text{low}=\text{mid}+1=8$

$\text{mid}=(\text{low}+\text{high})/2=8\text{-----}80 \quad 83>80 \quad \text{low}=\text{mid}+1=9$

$\text{mid}=(\text{low}+\text{high})/2=9\text{-----}150 \quad 83<125 \quad \text{high}=\text{mid}-1=8 \quad \text{high}<\text{low} \quad \text{return} -1$

查找失败

7. 有序表中关键字序列为：5, 10, 19, 21, 31, 37, 42, 48, 55, 150, 现要查找k为37及32的记录，写出其折半查找过程。

$\text{Low}=0 \quad \text{high}=9$

$\text{mid}=(\text{low}+\text{high})/2=4\text{-----}31 \quad 37>31 \quad \text{low}=\text{mid}+1=5$

$\text{mid}=(\text{low}+\text{high})/2=7\text{-----}48 \quad 37<48 \quad \text{high}=\text{mid}-1=6$

$\text{mid}=(\text{low}+\text{high})/2=5\text{-----}37$

查找成功

$\text{Low}=0 \quad \text{high}=9$

$\text{mid}=(\text{low}+\text{high})/2=4\text{-----}31 \quad 32>31 \quad \text{low}=\text{mid}+1=5$

$\text{mid}=(\text{low}+\text{high})/2=7\text{-----}48 \quad 32<48 \quad \text{high}=\text{mid}-1=6$

$\text{mid}=(\text{low}+\text{high})/2=5\text{-----}31 \quad 32>31 \quad \text{low}=\text{mid}+1=6$

$\text{mid}=(\text{low}+\text{high})/2=6\text{-----}42 \quad 32<42 \quad \text{high}=\text{mid}-1=5 \quad \text{high}<\text{low} \quad \text{return} -1$

查找失败

8. 设待排序的记录共7个，关键码分别为8, 3, 2, 5, 9, 1, 6。试用直接插入、直接选择两种方法，以关键码的变化描述排序全过程（动态过程），要求按递减顺序排序。

直接选择

8 3 2 5 9 1 6
 [1]3 2 5 9 8 6
 [1 2]3 5 9 8 6
 [1 2 3]5 9 8 6
 [1 2 3 5]9 8 6
 [1 2 3 5 6]8 9
 [1 2 3 5 6 8]9

直接插入

[8]3 2 5 9 1 6
 [8 3] 2 5 9 1 6
 [8 3 2] 5 9 1 6
 [8 5 3 2] 9 1 6
 [9 8 5 3 2] 1 6
 [9 8 5 3 2 1]6
 [9 8 6 5 3 2 1]

9. 设待排序文件共有12个记录，其关键字依次分别是28, 55, 06, 33, 161, 81, 91, 11, 25, 55, 57, 02, 请按选择排序的思想写出降序排序的全过程。

08 55 06 33 161 81 91 11 25 55 57 02
 [161]55 06 33 08 81 91 11 25 55 57 02
 [161 91]06 33 08 81 55 11 25 55 57 02
 [161 91 81]33 08 06 55 11 25 55 57 02
 [161 91 81 57]55 06 08 11 25 55 33 02
 [161 99 81 57 55]06 08 11 25 55 33 02
 [161 99 81 57 55 55]08 11 25 06 33 02
 [161 99 81 57 55 55 33]11 25 06 08 02
 [161 99 81 57 55 55 33 25]11 06 08 02
 [161 99 81 57 55 55 33 25 11]06 08 02
 [161 99 81 57 55 55 33 25 11 08]06 02
 [161 99 81 57 55 55 33 25 11 08 06]02

10. 设有待排序的8个数据记录，其排序用关键字的取值依次是14, 35, 18, 5, 7, 21, 35, 8, 请用简单选择排序法写出降序排序的每一趟结果。

14 35 18 5 7 21 35 8

[35]14 18 5 7 21 35 8

[35 35]18 5 7 21 14 8

[35 35 21]5 7 18 14 8

[35 35 21 18]7 5 14 8

[35 35 21 18 14]5 7 8

[35 35 21 18 14 8]7 5

[35 35 21 18 14 8 7]5

11. 设有待排序的8个数据记录，其排序用关键字的取值依次是68, 45, 20, 90, 15, 10, 50, 8, 请按直接选择排序的思想写出升序及降序排序的每一趟结果。

68 45 20 90 15 10 50 8

[90] 45 20 68 15 10 50 8

[90 68] 20 45 15 10 50 8

[90 68 50]45 15 10 20 8

[90 68 50 45] 15 10 20 8

[90 68 50 45 20] 10 15 8

[90 68 51 45 20 15]10 8

[90 68 51 45 20 15 10] 8

68 45 20 90 15 10 50 8

[8] 45 20 90 15 10 50 68

[8 10] 20 90 15 45 50 68

[8 10 15] 90 20 45 50 68

[8 10 15 20] 90 45 50 68

[8 10 15 20 45] 90 50 68

[8 10 15 20 45 50] 90 68

[8 10 15 20 45 50 68] 90

12. 假设待排序的一批记录的关键字序列为{14, 35, 18, 5, 7, 21}, 请给出按照简单选择排序方法依据关键字取值升序和降序两种情况下的排序过程。

13. 一个有序表的一批记录的关键字序列为 (7, 11, 15, 20, 32, 45, 63, 70, 82, 91), 存放在一个采用顺序存储结构表示的线性表中, 其中数组元素的下标为0, ..., 9的位置上分别对应存储有序表的第一元素直到最后一个元素。在折半查找中, 中间位置指示器的计算式子中采用取下整的方法, 请给出查找关键字值为82和关键字值为13的记录查找过程。

14. 某长度为10的有序表存储于一维结构数组中下标0~9的位置中, 其中记录的关键码值依次是: 5, 10, 18, 21, 33, 47, 48, 55, 80, 125, 现要查找关键码值为18及83的记录, 现规定在中间位置计算时采用“取下整”的方法, 写出折半查找的过程及查找结果。

这题型上边还有哑咩喋, 方法一样, 写起来太麻烦, 不写了思密达

15. 设有待排序的8个数据记录, 其排序用关键字的取值依次是68, 45, 20, 90, 15, 10, 50, 8, 请用冒泡排序法写出升序排序的每一趟结果。

这个是升序了.....不写了

16. 对于给定的一组关键字: 50, 38, 27, 16, 97, 76, 53, 66; 按关键字递减排序, 写出冒泡排序的各趟运行结果。

六、算法分析与设计题

1. 下面给出的算法的功能是在顺序存储结构表示的线性表中插入一个数据元素, 请画出算法的流程图, 在算法中对各个分支给出功能性注释。

```
#define Null 0
#define MaxSize 1024
typedef int DataType;
typedef struct node
{
```

```

    DataType data[MaxSize];
    int last;
}SequenList;

int Insert(SequenList *L,DataType x,int i)
{
    SequenList *p;
    int j;
    p=L;
    if(p->last==(MaxSize-1))
    {
        printf("\n线性表已经满了，无法再加入!");
        return Null;
    }
    else
        if( (i<1) || ( i>(p->last+1)))
        {
            printf("所给插入位置不在有效范围之内!");
            return Null;
        }
        else
        {
            for(j=L->last;j>=i-1;j--)
                L->data[j+1]=L->data[j];
            L->data[i-1]=x;
            L->last=L->last+1;
        }
        return (1);
}

```

1. 现要求完成同类型的两个线性表的合并运算，该运算将给定第二个线性表的元素追加到第一个线性表的最后一个元素之后，假定不会产生因空间不足而上溢的现象，运算完成后应修改第一个线性表的last分量以反映新的表长，运算不破坏第二个线性表。预期的功能用下面的计算示例及算法流程图表示。请将下列给定的算法设计填写完整。

算法功能描述：计算前 $L1=\{1,3,5,7,9\}$, $L2=\{2,4,6,8\}$

计算后： $L1=\{1,3,5,7,9,2,4,6,8\}$, $L2=\{2,4,6,8\}$

线性表的顺序存储结构定义：

```
#define MaxSize 1024  /*允许的最大数据元素数目*/

typedef int DataType; /*数据元素的类型*/

typedef struct  node{

        DataType data[MaxSize]; /*存储线性表中数据元素用的数组*/

        int last;    /*存储表的最后一个元素存放在data数组中的下标号*/

} SequeList;
```

计算前：

在L1表的存储结构体变量中，

```
data[0]=1,data[1]=3,data[2]=5,data[3]=7,data[4]=9

last=4;
```

在L2表的存储结构体变量中，计算前：

```
data[0]=2,data[1]=4,data[2]=6,data[3]=8

last=3
```

计算后：

在L1表的存储结构体变量中，

```
data[0]=1,data[1]=3,data[2]=5,data[3]=7,data[4]=9,data[5]=2,

data[6]=4,data[7]=6,data[8]=8

last=8;
```

在L2表的存储结构体变量中，计算前：

```
data[0]=2,data[1]=4,data[2]=6,data[3]=8

last=3
```

待完成的算法如下：

```
void MergeList(SequeList *L1,SequeList *L2)
```

{/*算法进入时：L1指向第一个线性表的结构体变量

L2指向第二个线性表的结构体变量

算法退出后：L1指向第一个线性表的结构体变量，其内容为合并后的表。

L2指向第二个线性表的结构体变量，其内容未变。

```
*/  
  
int i,j;    /*分别表示第一个表及第二个表的当前位置指示器*/  
  
int k;      /*循环控制变量*/  
  
int n1,n2; /*分别用于存放表L1及表L2的长度*/  
  
DataType x; /*存放从表L2中读出的数据元素*/  
  
n1=L1->last+1;  
n2=L2->last+1;  
i=L1->last+1;  
j=0;  
for(k=0;k< n2 ;k++)  
{  
    x= L2->data[j] ;  
    L1->data[i]=x;  
    i++;  
    j++;  
}  
  
    n1 =n1+n2-1;  
    return;  
}
```

算法流程图：

n

y

2. 阅读下面的程序，绘制出主函数main()的程序流程图，并说明本程序实现的功能。

```
#define maxsize 30
```

```
#define NULL -1
```

```
typedef char datatype;
```

```
typedef struct stack {
```

```
datatype elements[maxsize];
```

```
int top;
```

```
} S;
```

```
void IniStack(S *s) {
```

```
s->top=-1;
```

```
return;
```

```
}
```

```
int Push(S *s,datatype ch) {  
    if(s->top==maxsize-1) {  
        printf("\n stack overflow!\n");  
        return NULL; }  
    else {  
        s->top++;  
        s->elements[s->top]=ch;  
        return(1); }  
}
```

```
int Pop(S *s) {  
    if(s->top<0) {  
        printf("\nStack is empty!\n");  
        return NULL; }  
    else {  
        s->top--;  
        return(s->top+1); }  
}
```

```
main() {  
    char ch1;  
    S *s;  
    int f1=f2=1;  
    s=malloc(sizeof(S));  
    IniStack(s);  
    scanf("%c",&ch1);  
    while(ch1!='$') {  
        if(ch1=='(') f1=Push(s,ch1);  
        if(ch1==')') f2=Pop(s);  
        scanf("%c",&ch1); }  
}
```

```

if(s->top== -1&&f1!=-1&&f2!=-1) printf("\nBrackets match rightly\n");
else printf("\nThe expression is wrong.\n");
getch();
}

```

非线性数据结构部分：

一、填空题

1. 不考虑顺序的3个结点可构成 2 种不同形态的树， 5 种不同形态的二叉树。
2. 已知某棵完全二叉树的第4层有5个结点，则该完全二叉树叶子结点的总数为： 6 。
3. 已知一棵完全二叉树的第5层有3个结点，其叶子结点数是 9 。
4. 一棵具有110个结点的完全二叉树，若 $i=54$ ，则结点 i 的双亲编号是 27 ；结点 i 的左孩子结点的编号是 108 ，结点 i 的右孩子结点的编号是 109 。
5. 一棵具有48个结点的完全二叉树，若 $i=20$ ，则结点 i 的双亲编号是__10__；结点 i 的左孩子结点编号是__40__，右孩子结点编号是__41__。
6. 在有 n 个叶子结点的Huffman树中，总的结点数是： $2n-1$ 。
7. 图是一种非线性数据结构，它由两个集合 $V(G)$ 和 $E(G)$ 组成， $V(G)$ 是__顶点__的非空有限集合， $E(G)$ 是__边__的有限集合。
8. 遍历图的基本方法有 深度 优先搜索和 广度 优先搜索两种方法。
9. 图的遍历基本方法中 深度优先搜索 是一个递归过程。
10. n 个顶点的有向图最多有 $n(n-1)$ 条弧； n 个顶点的无向图最多有 $n(n-1)/2$ 条边。
11. 在二叉树的二叉链表中，判断某指针 p 所指结点是叶子结点的条件是 $p->lchild==NULL \&\& p->rchild==NULL$ 。
12. 在无向图 G 的邻接矩阵 A 中，若 $A[i,j]$ 等于1，则 $A[j,i]$ 等于 1 。

二、单项选择题

1. 树型结构的特点是：任意一个结点： (B)
A、可以有多个直接前趋 B、可以有多个直接后继
C、至少有1个前趋 D、只有一个后继
2. 如下图所示的4棵二叉树中， (C) 不是完全二叉树。

1. 深度为5的二叉树至多有 (C) 个结点。
- A、16 B、32 C、31 D、10
2. 64个结点的完全二叉树的深度为: (B)。
- A、8 B、7 C、6 D、5
3. 将一棵有100个结点的完全二叉树从根这一层开始, 每一层从左到右依次对结点进行编号, 根结点编号为1, 则编号为49的结点的左孩子的编号为: (A)。
- A、98 B、99 C、50 D、48
4. 在一个无向图中, 所有顶点的度之和等于边数的 (C) 倍。
- A、1/2 B、1 C、2 D、4
5. 设有13个值, 用它们组成一棵Huffman树, 则该Huffman树中共有 (D) 个结点。 $2n-1$
- A、13 B、12 C、26 D、25
6. 若对一棵有16个结点的完全二叉树按层编号, 则对于编号为7的结点x, 它的双亲结点及右孩子结点的编号分别为 (D)。
- A、2,14 B、2,15 C、3,14 D、3,15
7. 若对一棵有20个结点的完全二叉树按层编号, 则对于编号为5的结点x, 它的双亲结点及左孩子结点的编号分别为 (B)。
- A、2,11 B、2,10 C、3,9 D、3,10
8. 将一棵有100个结点的完全二叉树从根这一层开始, 每一层从左到右依次对结点进行编号, 根结点编号为1, 则编号最大的非叶结点的编号为: C
- A、48 B、49 C、50 D、51
9. 无向图的邻接矩阵是一个 (A)。
- A、对称矩阵 B、零矩阵 C、上三角矩阵 D、对角矩阵
10. 由64个结点构成的完全二叉树, 其深度为: (B)。
- A、8 B、7 C、6 D、5
11. 若对一棵有16个结点的完全二叉树按层编号, 则对于编号为7的结点x, 它的双亲结点及右孩子结点的编号分别为 (D)。
- A、2,14 B、2,15 C、3,14 D、3,15
16. 邻接表是图的一种 (B)。书上写“邻接表是一种顺序分配和链式分配相结合的储存结构”不确定A,B选哪个
- A、顺序存储结构 B、链式存储结构 C、索引存储结构 D、散列存储结构

1. 给定有向图如右图所示, 则该图的一个强连通分量是: (C)。

A、{A,B,C,F}

B、{B,C,F}

C、{B,C,D,F}

D、{C,D,E,F}

17. 已知一个有向图的邻接矩阵表示, 要删除所有从第*i*个结点发出的边, 应该: A

A、将邻接矩阵的第*i*行删除

B、将邻接矩阵的第*i*行元素全部置为0

C、将邻接矩阵的第*i*列删除

D、将邻接矩阵的第*i*列元素全部置为0

三、判断题VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV

1. (A) 非线性数据结构可以顺序存储, 也可以链接存储。

2. (B) 非线性数据结构只能用链接方式才能表示其中数据元素的相互关系。

3. (B) 完全二叉树一定是满二叉树。

4. (A) 在平衡二叉树中, 任意结点左右子树的高度差(绝对值)不超过1。

5. (B) 若一棵二叉树的任意一个非叶子结点的度为2, 则该二叉树为满二叉树。

6. (B) 度为1的有序树与度为1的二叉树是等价的。

7. (A) 二叉树的先序遍历序列中, 任意一个结点均排列在其孩子结点的前面。

8. (B) 已知一棵二叉树的先序序列和后序序列, 就一定能构造出该二叉树。

9. (B) 在霍夫曼树中, 权值最小的结点离根结点最近。

10. (B) 对任意一个图, 从它的某个顶点出发进行一次深度优先或广度优先遍历可访问到该图的每个顶点。

11. (B) 线性数据结构可以采用顺序存储结构或链式存储结构, 而非线性数据结构只能采用链式存储结构。

12. (A) 二叉树中的叶子结点就是二叉树没有左、右子树的结点。

13. (A) 如果一棵树中某结点的度为1, 则该结点仅有一棵子树。

14. (B) 在有向图中, 若存在有向边<V1,V2>, 则一定存在有向边<V2, V1>。

15. (A) 对任意一个图, 从它的某个顶点出发进行一次深度优先或广度优先遍历后, 并不一定能访问到该图的每个顶点。

16. (B) 用邻接矩阵法存储一个图时,在不考虑压缩存储的情况下,所占用的存储空间大小只与图中结点个数有关,而与图的边数无关。

四、简答题

1. 什么叫有序树? 什么叫无序树? 有序树和二叉树的差别是什么?

有序树定义(书上的): 树T中各子树 T_1, T_2, \dots, T_n 的相对次序是有意义的, 则称T为有序树

(网上的): 有序树和无序树 按照树中同层结点是否保持有序性, 可将树分为有序树和无序树。如果树中同层结点从左而右排列, 其次序不容互换, 这样的树称为有序树; 如果同层结点的次序任意, 这样的树称为无序树。

二叉树是每个结点最多有两个子树的有序树, 且二叉树有左右节点之分。有序树的子树虽然有相对次序, 但是只有一个子树的时候就没有相对次序了(即一个子树无法分左右)。

2. 什么叫完全二叉树? 什么叫满二叉树? 它们之间的关系是什么?

满二叉树: 如果一棵二叉树的深度为 k , 并且含有 $2^k - 1$ 个结点, 则称此二叉树为满二叉树。特点: 每一层上的结点数都是最大结点数

完全二叉树: 深度为 k , 有 n 个结点的二叉树, 当且仅当其每一个结点都与深度为 k 的满二叉树中的编号从1到 n 的节点一一对应时。称之为完全二叉树。特点: 叶子结点只可能在层次最大的两层上出现; 对任一结点, 若其右分支下子孙的最大层次为 l , 则其左分支下子孙的最大层次必为 l 或 $l+1$

3. 什么情况下二叉排序树的查找性能较好? 什么情况下二叉排序树的查找性能最差?

最好的情况是: 二叉排序树和二叉判定树形态相同。最坏的情况是: 二叉排序树为单支树, 这时的平均查找长度和顺序查找时相同。

五、综合题