

组编前言

依靠自己的力量,在有限的时间内学习一门新学科,从不懂到懂,从不会到会,从不理解到理解,从容易遗忘到记忆深刻,从不会应用到熟练应用,从模仿到创新,把书本知识内化为自己的知识,是一个艰难的过程。在这个过程中,自学者不仅需要认真钻研考试大纲,刻苦学习教材和辅导书,还应该做适量的练习,把学和练有机地结合起来,否则,就不能达到预定的学习目标。“纸上得来终觉浅,绝知此事要躬行。”这是每一位自学者都应遵循的信条。

编写练习,同样是件不容易的事。它对编写者提出了相当高的要求:

有较深的学术造诣;

有较丰富的教学经验;

对高等教育自学考试有深刻的理解并有一定的辅导自学者的经历;

对考试大纲、教材、辅导书有深入的了解,对文中的重点、难点、相互联系等有准确的理解;

对自学者学习需要和已有的知识基础有一定的了解。

只有把这些因素融会在一起,作者才能编写出高质量的,有利于举一反三、事半功倍的练习。

基于以上考虑,我们组织编写出版了同步练习册,使之与考试大纲、教材、自学辅导书相互补充,形成一个完整的学习媒体系统。

之所以把这些练习称为同步练习,是因为:

第一,它与考试大纲、教材的内容及顺序是一致的。按照考试大纲、教材的章、节、知识点的顺序编选习题,方便自学者循序渐进地学习与练习。

第二,它与自学者学习过程是一致的。自学过程大体包括初步接触、大体了解、理解、记忆、应用、创新、复习等阶段。在每一个阶段,自学者都容易找到相应的练习。

如此学与练同步的方式,有利于激发自学的兴趣与动机,有利于集中注意力于当前所学的内容,有利于理解、巩固、记忆、应用,尤其有利于自学者及时知道自己的学习状态与结果,以便随时调整学习计划,在难度较大处多投入精力。

基于对学习目标的考虑,我们把同步练习大致分为四类:

第一,单项练习:针对一个知识点而设计的练习。其目的在于帮助自学者理解和记忆基本概念和理论。

第二,综合练习:针对几个知识点而设的练习。这又可分为在本章综合、

跨章综合、跨学科综合三级水平。其目的在于帮助自学者把相关知识联系起来,形成特定的知识结构以便灵活地应用。

第三,创造性练习:提供一些案例、事实、材料,使考生应用所学到的理论、观点、方法创造性地解决问题。这类问题可能没有统一的答案,只有一些参考性的思路。其目的很明显,就是培养自学者的创新意识和能力。

第四,综合自测练习:在整个学科范围内设计练习,尽量参照考试大纲的题型,组成类似考卷的练习。其目的在于使自学者及时检测全部学习状况,帮助自学者作好迎接统一考试的知识及心理准备。

希望应考者在使用同步练习之前了解我们的构想,理解我们的意图,以便主动地选择适合自己学习的练习题目。

孔子说:“学而时习之,不亦乐乎。”一边学,一边练,有节奏、有规律地复习,不仅提高了学习效率,也会给艰难的学习过程带来不少的快乐。圣人能够体会到这一点,我们每一位自学者同样能体会到。如果通过这样的学习过程,实现了学习目标,实现了人生的理想,实现了对自我的不断超越,那么,我们说这种学习其乐无穷也毫不夸张。

全国高等教育自学考试指导委员会

2000年10月

目 录

第 1 部分 同步练习

第 1 章 概论	(3)
1.1 填空题	(3)
1.2 判断题	(3)
1.3 选择题	(3)
1.4 名词解释题	(3)
1.5 简答题	(4)
第 2 章 线性表	(6)
2.1 填空题	(6)
2.2 判断题	(6)
2.3 选择题	(6)
2.4 名词解释题	(7)
2.5 简答题	(7)
2.6 综合应用题	(7)
第 3 章 栈和队列	(10)
3.1 填空题	(10)
3.2 判断题	(10)
3.3 选择题	(10)
3.4 名词解释题	(11)
3.5 简答题	(11)
3.6 综合应用题	(11)
第 4 章 串	(14)
4.1 填空题	(14)
4.2 选择题	(14)
4.3 简答题	(14)
4.4 综合应用题	(14)
第 5 章 多维数组和广义表	(17)
5.1 填空题	(17)

5.2	判断题	(17)
5.3	选择题	(17)
5.4	名词解释题	(17)
5.5	简答题	(18)
5.6	综合应用题	(18)
第6章	树	(19)
6.1	填空题	(19)
6.2	判断题	(19)
6.3	选择题	(20)
6.4	简答题	(21)
6.5	综合应用题	(22)
第7章	图	(29)
7.1	填空题	(29)
7.2	判断题	(29)
7.3	选择题	(29)
7.4	名词解释题	(30)
7.5	简答题	(30)
7.6	综合应用题	(30)
第8章	排序	(36)
8.1	填空题	(36)
8.2	判断题	(36)
8.3	选择题	(36)
8.4	简答题	(37)
8.5	综合应用题	(38)
第9章	查找	(40)
9.1	填空题	(40)
9.2	判断题	(40)
9.3	选择题	(40)
9.4	名词解释题	(41)
9.5	简答题	(41)
9.6	综合应用题	(42)
第10章	文件	(45)
10.1	填空题	(45)
10.2	判断题	(45)
10.3	选择题	(45)
10.4	名词解释题	(45)
10.5	简答题	(45)
10.6	综合应用题	(46)

第 2 部分 参考答案

第 1 章 概论	(49)
1.1 填空题	(49)
1.2 判断题	(49)
1.3 选择题	(49)
1.4 名词解释题	(49)
1.5 简答题	(49)
第 2 章 线性表	(52)
2.1 填空题	(52)
2.2 判断题	(52)
2.3 选择题	(52)
2.4 名词解释题	(52)
2.5 简答题	(52)
2.6 综合应用题	(53)
第 3 章 栈和队列	(62)
3.1 填空题	(62)
3.2 判断题	(62)
3.3 选择题	(62)
3.4 名词解释题	(62)
3.5 简答题	(62)
3.6 综合应用题	(63)
第 4 章 串	(70)
4.1 填空题	(70)
4.2 选择题	(70)
4.3 简答题	(70)
4.4 综合应用题	(70)
第 5 章 多维数组和广义表	(76)
5.1 填空题	(76)
5.2 判断题	(76)
5.3 选择题	(76)
5.4 名词解释题	(76)
5.5 简答题	(76)
5.6 综合应用题	(77)
第 6 章 树	(80)
6.1 填空题	(80)
6.2 判断题	(80)

6.3 选择题	(80)
6.4 简答题	(80)
6.5 综合应用题	(83)
第7章 图	(98)
7.1 填空题	(98)
7.2 判断题	(98)
7.3 选择题	(98)
7.4 名词解释题	(98)
7.5 简答题	(98)
7.6 综合应用题	(99)
第8章 排序	(118)
8.1 填空题	(118)
8.2 判断题	(118)
8.3 选择题	(118)
8.4 简答题	(118)
8.5 综合应用题	(120)
第9章 查找	(130)
9.1 填空题	(130)
9.2 判断题	(130)
9.3 选择题	(130)
9.4 名词解释题	(130)
9.5 简答题	(130)
9.6 综合应用题	(132)
第10章 文件	(140)
10.1 填空题	(140)
10.2 判断题	(140)
10.3 选择题	(140)
10.4 名词解释题	(140)
10.5 简答题	(140)
10.6 综合应用题	(141)

附 录

模拟试卷	(143)
第1套	(143)
第2套	(147)
第3套	(151)
第4套	(157)

第 5 套.....	(160)
模拟试卷参考答案.....	(164)
第 1 套.....	(164)
第 2 套.....	(167)
第 3 套.....	(169)
第 4 套.....	(171)
第 5 套.....	(172)
 后 记.....	 (175)

第 1 部分 同步练习

第1章 概论

1.1 填空题

1. 算法的时间复杂度和_____合称为算法的复杂度。
2. 算法的时间复杂度不仅依赖于问题的规模,还与实例的_____有关。
3. 我们常常将数据的逻辑结构简称为_____,它有两大类:线性结构和_____。
4. 数据的存储结构有四种基本的存储方法:_____,链接存储方法、_____和散列存储方法。
5. _____是数据的基本单位。有些情况下它也称为元素、节点、顶点、记录。
6. 关于数据结构还没有一个标准定义,但它一般包括三个方面的内容:数据的_____,数据的_____和数据的运算。

1.2 判断题(判断下列各题是否正确,正确的在括号内打“√”,错误的打“×”)

1. 设三个函数 f, g, h 分别为: $f(n) = 100n^3 + n^2 + 1000$, $g(n) = 25n^3 + 5000n^2$, $h(n) = n^{1.5} + 5000n \lg n$.

请判断下列关系是否成立:

- (1) $f(n) = O(g(n))$
- (2) $g(n) = O(f(n))$
- (3) $h(n) = O(n^{1.5})$
- (4) $h(n) = O(n \lg n)$

2. 数据项是最基本的不可分割的数据单位,也是文件中可使用的数据最小单位。()
3. 数据的机内表示称为数据的存储结构。()

1.3 选择题

下列说法不正确的是()。

- A. 数据元素是数据的基本单位
- B. 数据项是数据中不可分割的最小标识单位
- C. 数据可由若干个数据元素构成
- D. 数据项可由若干个数据元素构成

1.4 名词解释题

数据——

数据元素——

数据类型——

数据结构——
 逻辑结构——
 存储结构——
 线性结构——
 非线性结构——

1.5 简答题

1. 举一个数据结构的例子,叙述其逻辑结构、存储结构、运算这三个方面的内容。
2. 常用的存储表示方法有哪几种?
3. 设有两个算法在同一机器上运行,其执行时间分别为 $100n^2$ 和 2^n ,要使前者快于后者, n 至少要多大?

4. 设 n 为正整数,利用“O”记号,将下列程序段的执行时间表示为 n 的函数。

```
(1) i=1; k=0;
    while(i<n){
        k=k+10*i; i++;
    }

(2) i=0; k=0;
    do{
        k=k+10*i; i++;
    } while(i<n);

(3) i=1; j=0;
    while(i+j<=n){
        if(i>j) j++;
        else i++;
    }

(4) x=n; // n>1
    while(x>=(y+1)*(y+1))
        y++;

(5) x=91; y=100;
    while(y>0)
        if(x>100){x=x-10; y--;}
        else x++;
```

5. 算法的时间复杂度仅与问题的规模相关吗?
6. 按增长率由小到大的顺序排列下列各函数:

2^{100} , $(3/2)^n$, $(2/3)^n$, n^n , $\sqrt{n}$, $n!$, 2^n , $\lg n$, $n^{\lg n}$, $n^{3/2}$

7. 有时为了比较两个同数量级算法的优劣,需突出主项的常数因子,而将低次项用“O”记号表示。例如, $T_1(n) = 1.39n \lg n + 100n + 256 = 1.39n \lg n + O(n)$, $T_2(n) = 2.0n \lg n - 2n = 2.0n \lg n + O(n)$,这两个式子表示,当 n 足够大时 $T_1(n)$ 优于 $T_2(n)$,因为前者的常数因子小于

后者。请用此方法表示下列函数,并指出当 n 足够大时,哪一个较优,哪一个较差?

$$(1) T_1(n) = 5n^2 - 3n + 60 \lg n$$

$$(2) T_2(n) = 3n^2 + 1000n + 3 \lg n$$

$$(3) T_3(n) = 8n^2 + 3 \lg n$$

$$(4) T_4(n) = 1.5n^2 + 6000n \lg n$$

第2章 线性表

2.1 填空题

1. 既无前趋又无后继的结点所在线性表的长度为_____, 结点指针域的值为_____。
2. 定义在线性表上的查找、读取、求表长和删除运算中, _____是加工型运算。
3. 设某双链表的结点形式为(prior, data, next), 若要在指针q所指结点(中间结点)的后面插入一个新结点s, 则需执行下述语句段: $s \rightarrow \text{prior} = q$; $s \rightarrow \text{next} = q \rightarrow \text{next}$; _____; $q \rightarrow \text{next} = s$;
4. 线性表 $L = (a_1, a_2, \dots, a_n)$ 采用顺序存储, 假定在不同的 $n+1$ 个位置上插入的概率相同, 则插入一个新元素平均需要移动的元素个数是_____。
5. 循环链表是一种首尾相连的链表。其特点是无须增加_____, 仅对表的_____稍作改变, 即可使表的处理更加方便灵活。
6. 顺序表的存储空间是_____分配的, 在程序执行之前必须明确规定它的存储规模。顺序表是由向量实现的, 它是一种_____存储结构。

2.2 判断题(判断下列各题是否正确, 正确的在括号内打“√”, 错误的打“×”)

1. 顺序存储线性表的存储空间是在对该线性表进行操作前就确定的。()
2. 链表是具有相同特性的数据元素的一个有限序列。()
3. 双向链表的最后一个结点的指针指向其前趋和表头结点, 就形成双循环链表。()
4. 线性表的链接存储, 表中元素的逻辑顺序与物理顺序一定相同。()
5. 对于频繁进行插入和删除的线性表, 宜采用链表作为存储结构。()
6. 当线性表的长度变化不大、易于事先确定其大小时, 宜采用顺序表作为存储结构。()

2.3 选择题

1. 栈和队列是两种特殊的线性表, 只能在它们的()添加或删除结点。
A. 中间点 B. 端点 C. 随机存取点 D. 结点
2. 在顺序存储的线性表 $(a_1, a_2, \dots, a_n)$ 中, 删除任意一个结点时所需移动结点的平均次数为()。
A. n B. $n/2$ C. $(n-1)/2$ D. $(n+1)/2$
3. 为了方便地在线性结构的数据中插入一个数据元素, 则其数据结构宜采用()方式。
A. 顺序存储 B. 链式存储 C. 索引存储 D. 散列存储

4. 在一个具有 n 个结点的双链表中插入一个新结点,则该操作的时间复杂性的量级为()。

- A. $O(1)$ B. $O(n)$ C. $O(\ln n)$ D. $O(n^2)$

5. 下列有关线性表的叙述中,正确的是()。

- A. 线性表中的元素之间的逻辑关系是线性的
B. 线性表中至少有一个元素
C. 线性表中任何一个元素有且仅有一个直接前趋
D. 线性表中任何一个元素有且仅有一个直接后继

6. 不用递归过程计算递归函数通常借助的数据结构是()。

- A. 线性表 B. 队列 C. 树 D. 栈

2.4 名词解释题

线性表——

顺序表——

存储密度——

2.5 简答题

1. 试描述头指针、头结点、开始结点的区别,并说明头指针和头结点的作用。

2. 何时选用顺序表、何时选用链表作为线性表的存储结构为宜?

3. 在顺序表中插入和删除一个结点需平均移动多少个结点?具体的移动次数取决于哪两个因素?

4. 为什么在单循环链表中设置尾指针比设置头指针更好?

5. 在单链表、双链表和单循环链表中,若仅知道指针 p 指向某结点,不知道头指针,能否将结点 $*p$ 从相应的链表中删去?若可以,其时间复杂度各为多少?

2.6 综合应用题

1. 下述算法的功能是什么?

```
typedef LinkList *ListNode;
LinkList Demo(LinkList L) { //L 是无头结点的单链表
    ListNode *Q, *P;
    if(L && L->next) {
        Q=L; L=L->next; P=L;
        while(P->next) P=P->next;
        P->next=Q; Q->next=NULL;
    }
    return L;
} //Demo
```

2. 设线性表的 n 个结点定义为 $(a_0, a_1, \dots, a_{n-1})$, 重写顺序表上实现的插入和删除算法:

InsertList 和 DeleteList

3. 试分别用顺序表和单链表作为存储结构,实现将线性表($a_0, a_1, \dots, a_{n-1}$)就地逆置的操作。所谓“就地”,是指辅助空间应为 $O(1)$ 。

4. 设顺序表 L 是一个递增有序表,试写一算法将 x 插入 L 中,并使 L 仍是一个有序表。

5. 设单链表 L 是一个递减有序表,试写一算法将 x 插入其中后仍保持 L 的有序性。

6. 写一算法在单链表上实现线性表的 $ListLength(L)$ 运算。

7. 已知 $head1$ 和 $head2$ 分别指向两个单链表的头结点,且已知其长度分别为 m 和 n 。试写一算法将这两个链表连接在一起,并分析该种算法的时间复杂度。

8. 设 A 和 B 是两个单链表,其表中元素递增有序。试写一算法将 A 和 B 归并成一个按元素值递减有序的单链表 C ,并要求辅助空间为 $O(1)$,请分析你的算法的时间复杂度。

9. 已知单链表 L 是一个递增有序表,试写一高效算法,删除表中值大于 min 且小于 max 的结点(若表中有这样的结点),同时释放被删结点的空间,这里 min 和 max 是两个给定的参数。并分析这一算法的时间复杂度。

10. 写一算法将单链表中值重复的结点删除,使所得的结果表中各结点值均不相同。

11. 假设在长度大于 1 的单循环链表中,既无头结点也无头指针, s 为指向链表中某个结点的指针,试编写算法删除结点 s 的直接前趋结点。

12. 已知由单链表表示的线性表中含有三类字符的数据元素(如字母字符、数字字符和其他字符),试编写算法构造三个以循环链表表示的线性表,使每个表中只含同一类的字符,且利用原表中的结点空间作为这三个表的结点空间,头结点可另辟空间。

13. 设有一个双链表,每个结点中除有 $prior$ 、 $data$ 和 $next$ 三个域外,还有一个访问频度域 $freq$,在链表被启用之前,其值均初始化为零。每当对链表进行一次 $LocateNode(L, x)$ 运算时,令元素值为 x 的结点中 $freq$ 域的值加 1,并调整表中结点的次序,使其按访问频度的递减序排列,以便使频繁访问的结点总是靠近表头。试写一符合上述要求的 $LocateNode$ 运算的算法。

14. 已知数组线性表类型定义如下:

```
#define maxlen 80
typedef struct {
    int elem[maxlen];
    int last;           // 线性表长度
}listtp;
listtp *list;
```

写一个算法,删除线性表中小于 0 的所有元素。

15. 单链表定义如下:

```
typedef struct list_node *list_pointer;
typedef struct list_node {
    char data[4];
    list_pointer link;
};
```

分析下面程序段的作用:

```
list_pointer function(list_pointer lead)
{
    list_pointer middle, trail;
    middle=NULL;
    while(lead)
    {
        trail=middle;
        middle=lead;
        lead=lead->link;
        middle->link=trail;
    }
    return middle;
}
```

16. 若用两个线性表 LA 和 LB 分别表示集合 A 和 B, 则求新的集合 $A=A \cup B$ 的算法如下, 试在横线上说明该行语句的作用。

```
void union(linear-list LA, linear-list LB);
{
    n=LENGTH(LA);
    for(i=1; i<=LENGTH(LB); i++)
    {
        x=GET(LB, i);
        k=LOCATE(LA, x);
        if(k==0)
        {
            INSERT(LA, n+1, x);
            n=n+1;
        }
    }
}
```

17. 对某带头结点的单链表的结点结构说明如下:

```
typedef struct node1
{
    int data;
    struct node1 * next;
} node;
```

试设计一个算法 int count(node * head, int m), 计算该单链表中数据域 data 的值为 m 的结点个数。设单链表的头指针为 head。

第3章 栈和队列

3.1 填空题

1. 设链队列 lq 中结点的格式为 $(data, next)$, 头指针为 $lq \rightarrow front$, 尾指针为 $lq \rightarrow rear$, 则队列为空的条件是_____。
2. 栈可看成是一种运算受限制的线性表, 其中可以进行插入和删除的一端称为_____。
3. 设栈 S 和队列 Q 的初始状态皆为空, 元素 a_1, a_2, a_3, a_4, a_5 和 a_6 依次通过一个栈, 一个元素出栈后即进入队列 Q , 若 6 个元素出队列的顺序是 $a_3, a_5, a_4, a_6, a_2, a_1$, 则栈 S 至少应该容纳_____个元素。

3.2 判断题(判断下列各题是否正确, 正确的在括号内打“√”, 错误的打“×”)

1. 栈和队列是两种重要的线性结构。()
2. 当栈满时再做进栈运算必定产生空间溢出, 简称“下溢”; 当栈空时再做退栈运算也会产生溢出, 简称“上溢”。()
3. 栈的链式存储结构称为“链栈”, 它是运算受限的单链表, 其插入和删除操作仅局限在表头位置进行。由于只能在表头部进行操作, 故链栈没有必要像单链表那样附加头结点。()

3.3 选择题

1. 栈是一种线性表, 它的特点是(从第 I 组中选择最佳答案)()。设用一维数组 $A[1..n]$ 来表示一个栈, $A[n]$ 为栈底。用整型变量 T 指示当前栈顶位置, $A[T]$ 为栈顶元素。往栈中推入(push)一个元素时, 变量 T 的值为(从第 II 组中选择最佳答案)(); 从栈中弹出(pop)一个元素时, 变量 T 的值为(从第 III 组中选择最佳答案)()。设栈空时, 有输入序列 a, b, c , 经过 push, pop, push, push, pop 操作后, 从栈中弹出的元素的序列是(从第 III 组中选择最佳答案)(), 变量 T 的值为(从第 IV 组中选择最佳答案)()。

I: A. 先进先出

B. 后进先出

C. 进优于出

D. 出优于进

E. 随机进出

II: A. 加 1

B. 减 1

C. 不变

D. 清 0

E. 加 2

F. 减 2

III: A. a, b B. b, c C. c, a D. b, a E. c, b F. a, c IV: A. $n+1$ B. $n+2$ C. n D. $n-1$ E. $n-2$