
注意：

1. 请独立完成，杜绝抄袭；
2. 提交时间：10-4。

第 2 章 线性表

1. 选择题

(1) 顺序表中第一个元素的存储地址是 100，每个元素的长度为 2，则第 5 个元素的地址是 ()。

- A. 110 B. 108 C. 100 D. 120

(2) 在 n 个结点的顺序表中，算法的时间复杂度是 $O(1)$ 的操作是 ()。

- A. 访问第 i 个结点 ($1 \leq i \leq n$) 和求第 i 个结点的直接前驱 ($2 \leq i \leq n$)
B. 在第 i 个结点后插入一个新结点 ($1 \leq i \leq n$)
C. 删除第 i 个结点 ($1 \leq i \leq n$)
D. 将 n 个结点从小到大排序

(3) 向一个有 127 个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动的元素个数为 ()。

- A. 8 B. 63.5 C. 63 D. 7

(4) 链接存储的存储结构所占存储空间 ()。

- A. 分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针
B. 只有一部分，存放结点值
C. 只有一部分，存放表示结点间关系的指针
D. 分两部分，一部分存放结点值，另一部分存放结点所占单元数

(5) 线性表若采用链式存储结构时，要求内存中可用存储单元的地址 ()。

- A. 必须是连续的 B. 部分地址必须是连续的
C. 一定是不连续的 D. 连续或不连续都可以

(6) 线性表 L 在 () 情况下适用于使用链式结构实现。

- A. 需经常修改 L 中的结点值 B. 需不断对 L 进行删除插入
C. L 中含有大量的结点 D. L 中结点结构复杂

-
- (7) 单链表的存储密度 ()。
- A. 大于 1 B. 等于 1 C. 小于 1 D. 不能确定
- (8) 将两个各有 n 个元素的有序表归并成一个有序表, 其最少的比较次数是 ()。
- A. n B. $2n-1$ C. $2n$ D. $n-1$
- (9) 在一个长度为 n 的顺序表中, 在第 i 个元素 ($1 \leq i \leq n+1$) 之前插入一个新元素时须向后移动 () 个元素。
- A. $n-i$ B. $n-i+1$ C. $n-i-1$ D. 1
- (10) 线性表 $L=(a_1, a_2, \dots, a_n)$, 下列说法正确的是 ()。
- A. 每个元素都有一个直接前驱和一个直接后继
B. 线性表中至少有一个元素
C. 表中诸元素的排列必须是由小到大或由大到小
D. 除第一个和最后一个元素外, 其余每个元素都有一个且仅有一个直接前驱和直接后继。
- (11) 创建一个包括 n 个结点的有序单链表的时间复杂度是 ()。
- A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n \log_2 n)$
- (12) 以下说法错误的是 ()。
- A. 求表长、定位这两种运算在采用顺序存储结构时实现的效率不比采用链式存储结构时实现的效率低
B. 顺序存储的线性表可以随机存取
C. 由于顺序存储要求连续的存储区域, 所以在存储管理上不够灵活
D. 线性表的链式存储结构优于顺序存储结构
- (13) 在单链表中, 要将 s 所指结点插入到 p 所指结点之后, 其语句应为 ()。
- A. $s \rightarrow \text{next} = p+1; p \rightarrow \text{next} = s;$
B. $(*p).\text{next} = s; (*s).\text{next} = (*p).\text{next};$
C. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s \rightarrow \text{next};$
D. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
- (14) 在双向链表存储结构中, 删除 p 所指的结点时须修改指针 ()。
- A. $p \rightarrow \text{next} \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} \rightarrow \text{next} = p \rightarrow \text{next};$
B. $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{prior} = p;$
C. $p \rightarrow \text{prior} \rightarrow \text{next} = p; p \rightarrow \text{prior} = p \rightarrow \text{prior} \rightarrow \text{prior};$
D. $p \rightarrow \text{prior} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{next} = p \rightarrow \text{prior} \rightarrow \text{prior};$
- (15) 在双向循环链表中, 在 p 指针所指的结点后插入 q 所指向的新结点, 其修改指针的操作是 ()。
- A. $p \rightarrow \text{next} = q; q \rightarrow \text{prior} = p; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{next} = q;$
B. $p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next};$
C. $q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{prior} = q; p \rightarrow \text{next} = q;$
D. $q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q;$

2. 算法设计题 (请进行算法分析, 并写出相应的函数代码)

(1) 将两个递增的有序链表合并为一个递增的有序链表。要求结果链表仍使用原来两个链表的存储空间, 不另外占用其它的存储空间。表中不允许有重复的数据。

如:

```
void MergeList(LinkList &La, LinkList &Lb, LinkList &Lc)
{
    .....
}
```

(2) 将两个非递减的有序链表合并为一个非递增的有序链表。要求结果链表仍使用原来两个链表的存储空间, 不另外占用其它的存储空间。表中允许有重复的数据。

(3) 已知两个链表 A 和 B 分别表示两个集合, 其元素递增排列。请设计算法求出 A 与 B 的交集, 并存放于 A 链表中。

(4) 已知两个链表 A 和 B 分别表示两个集合, 其元素递增排列。请设计算法求出两个集合 A 和 B 的差集 (即仅由在 A 中出现而不在 B 中出现的元素所构成的集合), 并以同样的形式存储, 同时返回该集合的元素个数。

(5) 设计算法将一个带头结点的单链表 A 分解为两个具有相同结构的链表 B、C, 其中 B 表的结点为 A 表中值小于零的结点, 而 C 表的结点为 A 表中值大于零的结点 (链表 A 中的元素为非零整数, 要求 B、C 表利用 A 表的结点)。

(6) 设计一个算法, 通过一趟遍历在单链表中确定值最大的结点。

(7) 设计一个算法, 通过遍历一趟, 将链表中所有结点的链接方向逆转, 仍利用原表的存储空间。

(8) 设计一个算法, 删除递增有序链表中值大于 mink 且小于 maxk 的所有元素 (mink 和 maxk 是给定的两个参数, 其值可以和表中的元素相同, 也可以不同)。

(9) 已知 p 指向双向循环链表中的一个结点, 其结点结构为 data 、 prior 、 next 三个域, 写出算法 $\text{change}(p)$, 交换 p 所指向的结点和它的前缀结点的顺序。

(10) 已知长度为 n 的线性表 A 采用顺序存储结构, 请写一时间复杂度为 $O(n)$ 、空间复杂度为 $O(1)$ 的算法, 该算法删除线性表中所有值为 item 的数据元素。