

## 第2章 线性表

1650254 尤尧寅 计算机一班

- 1、设线性表有  $n$  个元素，以下操作中，\_\_A\_\_在顺序表上实现比在链表上实现效率更高
  - A 输出第  $i$  个元素值 ( $i$  在  $1-n$  之间)
  - B 交换第 1 个元素与第 2 个元素的值
  - C 顺序输出这  $n$  个元素的值
  - D 输出与给定值  $x$  相等的元素在线性表中的序号
- 2、设线性表中有  $2n$  个元素，以下操作中，\_\_A\_\_在单链表上实现要比在顺序表上实现效率更高
  - A 删除指定的元素
  - B 在最后一个元素的后面插入一个新元素
  - C 顺序输出前  $k$  个元素
  - D 交换第  $i$  个元素和第  $2n-i-1$  个元素的值 ( $i$  在  $0 - n-1$  间)
- 3、如果最常用的操作是取第  $i$  个结点及其前驱，则采用\_\_D\_存储方式最节省时间
  - A 单链表
  - B 双链表
  - C 单循环链表
  - D 顺序表
- 4、将两个各有  $n$  个元素的有序顺序表(某个表中的元素，两个表之间的元素，值均有可能相同)归并成一个有序顺序表，其最少比较次数是\_\_A\_\_
  - A  $n$
  - B  $2n-1$
  - C  $2n$
  - D  $n-1$
- 5、在一个长度为  $n(n>1)$  的带头结点单链表  $h$  上，另设有尾指针  $r$  (指向尾结点)，执行\_\_B\_\_的操作与链表的长度有关
  - A 删除单链表中的第一个元素
  - B 删除单链表的最后一个元素
  - C 在单链表的第一个元素前插入一个新元素
  - D 在单链表的最后一个元素后插入一个新元素
- 6、双向循环链表中，在  $p$  结点之前插入  $q$  结点的操作是\_\_D\_\_
  - A  $p \rightarrow \text{prior} = q;$   
 $q \rightarrow \text{next} = p;$   
 $p \rightarrow \text{prior} \rightarrow \text{next} = q;$   
 $q \rightarrow \text{prior} = p \rightarrow \text{prior};$
  - B  $p \rightarrow \text{prior} = q;$   
 $p \rightarrow \text{prior} \rightarrow \text{next} = q;$   
 $q \rightarrow \text{next} = p;$   
 $q \rightarrow \text{prior} = p \rightarrow \text{prior};$
  - C  $q \rightarrow \text{next} = p;$   
 $q \rightarrow \text{prior} = p \rightarrow \text{prior};$   
 $p \rightarrow \text{prior} = q;$   
 $p \rightarrow \text{prior} \rightarrow \text{next} = q;$

```

D q->next=p;
  q->prior=p->prior;
  p->prior->next=q;
  p->prior=q;

```

7、在一个单链表中删除 p 结点(假设 p 不是尾结点)时, 应执行如下操作:

- (1) q=p->next;
- (2) p->data=p->next->data;
- (3) p->next=\_\_\_\_\_ p->next->next \_\_\_\_\_;
- (4) free(q);

8、在一个单链表中的 p 结点之前插入一个 s 结点, 可执行如下操作:

- (1) s->next=\_\_\_\_\_ p->next \_\_\_\_\_
- (2) p->next=s;
- (3) t=p->data;
- (4) p->data=\_\_\_\_\_ s->data \_\_\_\_\_
- (5) s->data=\_\_\_\_\_ t \_\_\_\_\_

9、在一个双向循环链表中删除 p 结点时, 应执行如下操作:

- (1) \_\_\_\_\_ p->next->prior \_\_\_\_\_ = p->prior;
- (2) p->prior->next = \_\_\_\_\_ p->next \_\_\_\_\_;
- (3) free(p);

10、在单链表、双向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 p 从相应的链表中删除(不允许进行结点之间数据域的复制)? 若可以, 时间复杂度各为多少?

单链表:

在单链表中, 不能将点 p 从相应链表中删除

双向链表:

在双向链表中, 可以将点 p 从相应链表中删除

```
p->prior->next=p->next;
```

```
p->next->prior=p->prior;
```

时间复杂度为  $O(1)$

单循环链表

在单循环链表中, 可以将点 p 从相应链表中删除

```
q=p;
```

```
while(q->next!=p)
```

```
    q=q->next;
```

时间复杂度为  $O(n)$

11、设计一个高效算法, 将顺序表的所有元素逆置, 要求算法的空间复杂度为  $O(1)$

```

{
    elemnt_type temp;
    for (int i = 0; i < p->length; i++)
    {
        temp = p->ele[i];
        p->ele[i] = p->ele[p->length - 1 - i];
        p->ele[p->length - 1 - i] = temp;
    }
}

```

```
}
```

12、设计一个高效算法，从顺序表中删除所有元素值为 x 的元素，要求空间复杂度为  $O(1)$

```
{
    int i = p->length - 1;
    for (; i >= 0; i--)
    {
        if (p->ele[i] == x)
        {
            for (int j = i; j < p->length - 2; j++)
                p->ele[j] = p->ele[j + 1];
            p->length - 1;
        }
    }
}
```

13、用顺序表表示集合，设计一个求集合交集的算法

```
{
    int num = 1;
    for(int i=0; i<A->length; i++)
        for(int j=0; j<B->length; j++)
            if (A->ele[i] == B->ele[j])
            {
                ListInsert(C, num++, A->ele[i]);
            }
}
```

14、从头结点的循环单链表中删除值为 x 的第一个结点

```
{
    linklist *p = *head;
    if (head == NULL)
        return error;
    while (p)
    {
        if(p->next==NULL)
            break;
        if (p->next->data == x)
        {
            linklist *q = p->next;
            p->next = p->next->next;
            free(p);
            break;
        }
        p = p->next;
    }
}
```

- 15、 假定有一个带附加表头结点的链接表，表头指针为 HL，每个结点含三个域:data, next 和 range，其中 data 为值域，next 和 range 均为指针域，现在所有结点已经由 next 域链接起来，试编一算法，利用 range 域(此域的初始值均为 NULL)把所有结点按照其值从小到大的顺序链接起来

```
{
    linklist *p = *hl;
    linklist head_range;
    linklist *q = head_range;
    linklist *temp;
    element_type min1 = p->next->data;
    element_type min2 = p->next->data;
    bool flag = 0;
    while (1)
    {
        flag = 0;
        min1 = min2;
        while (p)
        {
            if (p->data <= min1 && p->range == NULL)
            {
                temp = p;
                min2 = min1;
                min1 = p->data;
                flag = 1;
            }
            p = p->next;
        }
        if (flag == 0)
            break;
        if (head_range->range == NULL)
            head_range->range = temp;
        q->range = temp;
        q = q->range;
    }
}
```

- 16、 已知带头结点的单链表 L 是一个递增有序表，设计一个高效算法，删除表中 data 值在 [min .. max] 之间的所有结点，并分析算法的时间复杂度

```
{
    linklist *p = *L;
    while (p->next != NULL)
    {
        if (p->next->data > min)
        {
            linklist *temp_p = *p;
            while (p->next != NULL)
```

```

{
    if (p->next->data > max)
    {
        temp_p->next = p->next;
        free(p);
        break;
    }
    else
    {
        linklist *q = p;
        p = p->next;
        free(q);
    }
}
break;
}
p = p->next;
}
}

```

17、 有一个值按非递减有序排列的单链表，设计一个算法删除值域重复的结点，并分析算法的时间复杂度

```

{
    linklist *p = *head;
    if (head == NULL)
        return ERROR;
    while (p->next != NULL)
        if (p->next->data == p->data)
        {
            linklist *q ;
            q=p->next;
            p->next = q->next;
            free(q);
        }
        else
            p = p->next;
}
}

```

18、 用单链表表示集合，设计一个算法表示集合的交

```

{
    linklist *p;
    p=A->next;
    int num = 1;
    while (p)
    {
        linklist *q;
        q== B->next;
    }
}

```

```

    while (q)
    {
        if (p->data == q->data)
            ListInsert(C, num++, A->ele[i]);
        q = q->next;
    }
    p = p->next;
}
}

```

19、 写出将带头结点的双向循环链表倒置的算法

```

{
    linklist *p;
    p=head->next;
    while (p!= head)
    {
        linklist *q ;
        q= p->next;
        linklist *temp_p;
        temp_p= p->next;
        p->next = p->piror;
        p->piror = temp_p;
        p = q;
    }
}

```

20、 设有一个双向链表 h, 设计一个算法查找第一个元素值为 x 的结点，将其与后继结点进行交换

```

{
    linklist *p = *head;
    while (p->next != NULL)
    {
        if (p->data == x)
        {
            element_type s = p->data;
            p->data = p->next->data;
            p->next->data = s;
            break;
        }
        p = p->next;
    }
}

```

### 【作业要求:】

1、 **5月3日前**网上提交本次作业（直接在本文件中作答，转换为 PDF 后提交即可）

- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明
- 4、**答案用其它颜色标注**