

数据结构习题课

主要内容

- 线性表

- 栈和队列

- 串

- 数组和广义表

- 树和二叉树

- 图

- 查找

- 排序

线性表-主要内容

- 线性表的类型定义
- 线性表的顺序表示和实现
- 线性表的链式表示和实现
- 一元多项式的处理

线性表-习题

1. 在 n 个元素的线性表的数组表示中，时间复杂度为 $O(1)$ 的操作是()。

I. 访问第 i ($1 \leq i \leq n$)个结点和求第 i ($2 \leq i \leq n$)个结点的直接前驱

II. 在最后一个结点后插入一个新的结点

III. 删除第1个结点

IV. 在第 i ($1 \leq i \leq n$)个结点后插入一个结点

A. I

B. II、 III

C. I、 II

D. I、 II、 III

2. 顺序表的插入算法中，当 n 个空间已满时，可再申请增加分配 m 个空间，若申请失败，则说明系统没有()可分配的存储空间

A. m 个

B. m 个连续

C. $n+m$ 个

D. $n+m$ 个连续

线性表-习题

1. 在 n 个元素的线性表的数组表示中，时间复杂度为 $O(1)$ 的操作是(C)。

I. 访问第 i ($1 \leq i \leq n$)个结点和求第 i ($2 \leq i \leq n$)个结点的直接前驱

II. 在最后一个结点后插入一个新的结点

III. 删除第1个结点

IV. 在第 i ($1 \leq i \leq n$)个结点后插入一个结点

A. I

B. II、 III

C. I、 II

D. I、 II、 III

II中, 在最后位置插入新结点不需要移动元素; III和IV中, 被删/插入结点后的结点需依次前移, 时间复杂度为 $O(n)$ 。

2. 顺序表的插入算法中，当 n 个空间已满时，可再申请增加分配 m 个空间，若申请失败，则说明系统没有(D)可分配的存储空间

A. m 个

B. m 个连续

C. $n+m$ 个

D. $n+m$ 个连续

线性表-习题

3. 在双链表中向P所指的结点之前插入一个结点q的操作为()。

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. $q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p; p \rightarrow \text{prior} = q \rightarrow \text{next};$

C. $q \rightarrow \text{next} = p; p \rightarrow \text{next} = q; q \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p;$

D. $p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p; q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} = q;$

4. 某线性表用带头结点的循环单链表存储，头指针为head, 当 $\text{head} \rightarrow \text{next} \rightarrow \text{next} = \text{head}$ 成立时，线性表长度可能是()。

A. 0

B. 1

C. 2

D. 可能为0或1

线性表-习题

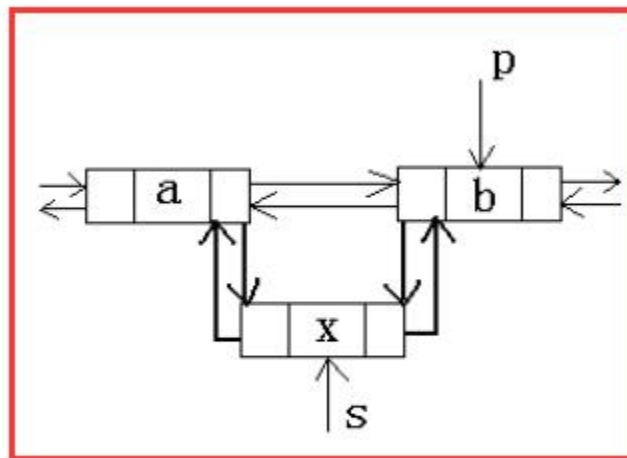
3. 在双链表中向P所指的结

A. $p \rightarrow \text{prior} = q; q \rightarrow \text{next} \rightarrow \text{prior} = p;$

B. $q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} = q \rightarrow \text{next};$

C. $q \rightarrow \text{next} = p; p \rightarrow \text{next} = q;$ 在P结点之前插入S结点前后的指针变化 $t = p;$

D. $p \rightarrow \text{prior} \rightarrow \text{next} = q; q \rightarrow \text{next} = p; q \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} = q;$



作为(D)。

$\text{or} = p \rightarrow$

$p; p \rightarrow$

4. 某线性表用带头结点的循环单链表存储，头指针为head, 当 $\text{head} \rightarrow \text{next} \rightarrow \text{next} = \text{head}$ 成立时，线性表长度可能是(D)。

A. 0

B. 1

C. 2

D. 可能为0或1

对空循环链表，头节点的next还是指向本身；含有一个元素的循环链表，头节点的next域指向唯一元素，其next域指向头节点。

线性表-习题

5. 在一个长度为 n 的带头结点的单链表 h 上，设有尾指针 r ，则执行（ ）操作与链表的表长有关。

A. 删除单链表中的第一个元素

B. 删除单链表中最后一个元素

C. 在单链表第一个元素前插入一个新元素

D. 在单链表最后一个元素后插入一个新元素

线性表-习题

5. 在一个长度为 n 的带头结点的单链表上，设有尾指针 r ，则执行（
B）操作与链表的表长有关。

A. 删除单链表中的第一个元素

B. 删除单链表中最后一个元素

C. 在单链表第一个元素前插入一个新元素

D. 在单链表最后一个元素后插入一个新元素

删除单链表的最后一个节点需要尾结点的前驱指针。

线性表-习题

将两个有序顺序表合并成一个新的有序顺序表，并由函数返回结果顺序表。

算法思想:首先，按顺序不断取下两个顺序表表头较小的结点存入新的顺序表中。然后，看哪个表还有剩余，将剩下的部分加到新的顺序表后面。

线性表-习题

算法思想:首先,按顺序不断取下两个顺序表表头较小的结点存入新的顺序表中。然后,看哪个表还有剩余,将剩下的部分加到新的顺序表后面。

```
本题代码如下: bool Merge(SeqList A, SeqList B, SeqList &C) {  
    //合并有序顺序表 A 与 B 成为一个新的有序顺序表 C  
    if (A.length+B.length>C.maxSize)    //大于顺序表的最大长度  
        return false;  
    int i=0, j=0, k=0;  
    while(i<A.length&& j<B.length) {    //循环, 两两比较, 小者存入结果表  
        if (A.data[i]<=B.data[j])  
            C.data[k++]=A.data[i++];  
        else  
            C.data[k++]=B.data[j++];  
    }  
    while(i<A.length)    //还剩一个没有比较完的顺序表  
        C.data[k++]=A.data[i++];  
    while(j<B.length)  
        C.data[k++]=B.data[j++];  
    C.length=k+1;  
    return true;  
}
```

主要内容

- 线性表

- 栈和队列

- 串

- 数组和广义表

- 树和二叉树

- 图

- 查找

- 排序

栈和队列-主要内容

- 栈
- 栈的应用举例
- 栈与递归的实现
- 队列

栈和队列-习题

1. 假定利用数组 $a[n]$ 顺序存储一个栈，用 top 表示栈顶指针，用 $top == -1$ 表示栈空，并已知栈未满，当元素 x 进栈时所执行的操作为()。

A. $a[--top] = x$

B. $a[top--] = x$

C. $a[++top] = x$

D. $a[top++] = x$

2. 设有一个顺序共享栈 $Share[0:n-1]$ ，其中第一个栈顶指针 $top1$ 的初值为 -1 ，第二个栈顶指针 $top2$ 的初值为 n ，则判断共享栈满的条件是()。

A. $top2 - top1 == 1$

B. $top1 - top2 == 1$

C. $top1 == top2$

D. 以上都不对

3. 设有一个顺序共享栈 $Share[0:n-1]$ ，其中第一个栈顶指针 $top1$ 的初值为 -1 ，第二个栈顶指针 $top2$ 的初值为 $n-1$ ，则判断共享栈满的条件是()。

栈和队列-习题

1. 假定利用数组 $a[n]$ 顺序存储一个栈，用 top 表示栈顶指针，用 $top == -1$ 表示栈空，并已知栈未满，当元素 x 进栈时所执行的操作为(C)。

A. $a[--top] = x$

B. $a[top--] = x$

C. $a[++top] = x$

D. $a[top++] = x$

初始时 top 为负，入栈后 $top=0$, 所以应先自增

栈和队列-习题

2. 设有一个顺序共享栈Share[0:n-1]，其中第一个栈顶指针top1的初值为-1，第二个栈顶指针top2的初值为n，则判断共享栈满的条件是(A)。

A. $top2 - top1 == 1$

B. $top1 - top2 == 1$

C. $top1 == top2$

D. 以上都不对

3. 设有一个顺序共享栈Share[0:n-1]，其中第一个栈顶指针top1的初值为-1，第二个栈顶指针top2的初值为n-1，则判断共享栈满的条件是(C)。

栈顶指针与对头队尾的指针是不固定的，要注意审题

栈和队列-习题

4. 已知循环队列的存储空间为数组A[21], front指向队头元素的前一个位置, rear指向队尾元素, 假设当前front和rear的值分别为8和3, 则该队列的长度为()。

- A. 5
- B. 6
- C. 16
- D. 17

5. 在一个链队列中, 假设队头指针为front, 队尾指针为rear, x所指向的元素需要入队, 则需要执行的操作为()。

- A. front=x, front= front->next
- B. x->next= front->next, front=x
- C. rear->next=x, rear=x
- D. rear->next=x, x->next=null, rear=x .

栈和队列-习题

4. 已知循环队列的存储空间为数组A[21], front指向队头元素的前一个位置, rear指向队尾元素, 假设当前front和rear的值分别为8和3, 则该队列的长度为(C)。

- A. 5
- B. 6
- C. 16
- D. 17

$\text{Rear} < \text{front}, (\text{rear} - \text{front} + \text{maxsize}) \% \text{maxsize} = 16$

5. 在一个链队列中, 假设队头指针为front, 队尾指针为rear, x所指向的元素需要入队, 则需要执行的操作为(D)。

- A. $\text{front} = \text{x}, \text{front} = \text{front} \rightarrow \text{next}$
- B. $\text{x} \rightarrow \text{next} = \text{front} \rightarrow \text{next}, \text{front} = \text{x}$
- C. $\text{rear} \rightarrow \text{next} = \text{x}, \text{rear} = \text{x}$
- D. $\text{rear} \rightarrow \text{next} = \text{x}, \text{x} \rightarrow \text{next} = \text{null}, \text{rear} = \text{x}$

C不够严谨, 注意将队尾元素的next域置为空。

栈和队列-习题

6. 假设循环单链表表示的队列长度为 n , 队头固定在链表表尾, 若只设头指针, 则进队操作的时间复杂度为()。

A. $O(n)$

B. $O(1)$

C. $O(n^2)$

D. $O(n \log_2(n))$

A 进队是从队尾进, 需要遍历一遍单链表

主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找
- 排序

串-主要内容

- 串的基本操作的定义，并能利用这些基本操作来实现串的其它各种操作。
- 串的各种存储结构的特点及其适用场合。
- 串模式匹配算法。
- KMP算法，next函数和nextval函数值。

串-习题

- 1、设有两个串S1和S2，求S2在S1中首次出现的位置的运算称作()
 - A. 求子串 B. 判断是否相等
 - C. 模式匹配 D. 连接
- 2、KMP算法的特点是模式匹配时指示主串的指针（）
 - A. 不会变大 B. 不会变小
 - C. 都有可能 D. 无法判断
- 3、设主串的长度为n，子串的长度为m，则简单的模式匹配算法的时间复杂度为（），KMP算法的时间复杂度为（）。
 - A. $O(m)$ B. $O(n)$ C. $O(mn)$ D. $O(m+n)$

串-习题

- 1、设有两个串S1和S2，求S2在S1中首次出现的位置的运算称作()
 - A. 求子串
 - B. 判断是否相等
 - C. 模式匹配
 - D. 连接
- 2、KMP算法的特点是模式匹配时指示主串的指针（）
 - A. 不会变大
 - B. 不会变小(KMP算法主串不会回溯)
 - C. 都有可能
 - D. 无法判断
- 3、设主串的长度为n，子串的长度为m，则简单的模式匹配算法的时间复杂度为（C），KMP算法的时间复杂度为（D）。
 - A. $O(m)$
 - B. $O(n)$
 - C. $O(mn)$ （理论上）
 - D. $O(m+n)$

串-习题

- 4、已知串 $S = 'aaab'$ ，其next数组值为（ ）。
A. 0123 B. 0112 C. 0231 D. 1211
- 5、已知字符串S为'abaabaabacacaabaabcc'，模式串t为'abaabc'。采用KMP算法进行匹配，第一次出现“失败”（ $s[i] \neq t[j]$ ）时， $i=j=5$ ，则下次开始匹配时，i和j的值分别是（ ）。
A. $i=1, j=0$ B. $i=5, j=0$
C. $i=5, j=2$ D. $i=6, j=2$

串-习题

■ 4、已知串 $S = 'aaab'$ ，其next数组值为（）。

A. 0123 B. 0112 C. 0231 D. 1211

1) 设 $next[1]=0$, $next[2]=1$ 。

2) $j=3$ 时 $k=next[j-1]=next[2]=1$ ，观察 $s[j-1]$ ($s[2]$) 与 $S[k]$ ($S[1]$) 是否相等，

$S[2]=a$, $S[1]=a$, $S[2]=S[1]$ ，所以 $next[j]=k+1=2$

3) $j=4$ 时 $k=next[j-1]=next[3]=2$ ，观察 $S[j-1]$ ($S[3]$) 与 $S[k]$ ($S[2]$) 是否相等， $S[3]=a$, $S[2]=a$, $s[3]=S[2]$ ，所以 $next[j]=k+1=3$ 。

串-习题

- 5、已知字符串S为'abaabaabacacaabaabcc'，模式串t为'abaabc'。采用KMP算法进行匹配，第一次出现“失败”（ $s[i] \neq t[j]$ ）时， $i=j=5$ ，则下次开始匹配时， i 和 j 的值分别是（ ）。

- A. $i=1, j=0$ B. $i=5, j=0$
C. $i=5, j=2$ D. $i=6, j=2$

编号	0	1	2	3	4	5
t	a	b	a	a	b	c
next	-1	0	0	1	1	2

发生失配时，主串指针 i 不变，子串指针 j 回退到 $next[j]$ 位置重新比较，当 $s[i]=t[j]$ 时， $i=j=5$ ，由 $next$ 表得知 $next[j]=next[5]=2$ （位序从0开始）。因此， $i=5$ 、 $j=2$ 。

主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找
- 排序

数组和广义表-主要内容

- 数组类型的特点以及在高级编程语言中的两种存储表示和实现方法。
- 指定下标的元素在存储结构中的地址计算方法。
- 掌握矩阵压缩存储的常用方法。
- 掌握广义表的结构特点和表长、表深、表头、表尾的定义。

数组和广义表-主要内容

- 广义表的存储表示方法。
- 对非空广义表进行分解的两种分析方法：即可将一个非空广义表分解为表头和表尾两部分或者分解为 n 个子表。
- 巩固递归算法的设计思想。

数组和广义表-习题

- 1、对特殊矩阵采用压缩存储的主要目的是（ ）。
 - A. 表达变得简单 B. 对矩阵元素的存取变得简单
 - C. 去掉矩阵中的多余元素 D. 减少不必要的存储空间
- 2、对 n 阶对称矩阵压缩存储时，需要表长为（ ）的顺序表。
 - A. $n/2$ B. $n \times n/2$ C. $n(n+1)/2$ D. $n(n-1)/2$
- 3、有一个 $n \times n$ 的对称矩阵 A ，将其下三角部分按行存放在一维数组 B 中，而 $A[0][0]$ 存放于 $B[0]$ 中，则第 i 行的对角元素 $A[i][i]$ 存放于 B 中的（ ）处。
 - A. $(i+3)i/2$ B. $(i+1)i/2$
 - C. $(2n-i+1)i/2$ D. $(2n-i-1)i/2$

数组和广义表-习题

- 1、对特殊矩阵采用压缩存储的主要目的是（ ）。
 - A. 表达变得简单 B. 对矩阵元素的存取变得简单
 - C. 去掉矩阵中的多余元素 **D. 减少不必要的存储空间**

- 2、对n阶对称矩阵压缩存储时，需要表长为（ ）的顺序表。
 - A. $n/2$ B. $n \times n/2$ **C. $n(n+1)/2$** D. $n(n-1)/2$

只需存储其上三角或下三角部分（含对角线），元素个数为 $n+(n-1)+(n-2)+\dots+1=n(n+1)/2$ 。

数组和广义表-习题

- 3、有一个 $n \times n$ 的对称矩阵A，将其下三角部分按行存放在一维数组B中，而A[0][0]存放于B[0]中，则第i行的对角元素A[i][i]存放于B中的（ ）处。

A. $(i+3)i/2$ B. $(i+1)i/2$
C. $(2n-i+1)i/2$ D. $(2n-i-1)i/2$

此题要注意3个细节:矩阵的最小下标为0;数组下标也是从0开始的;矩阵按行优先存在数组中。注意到此三点，答案不难得到为A。此外，本类题建议采用特殊值代入法求解，例如，A[1][1]对应的下标应为2，代入后只有A满足条件。

技巧:对于特殊三角矩阵压缩存储的题，心中应有“平移”搬动的思想，并结合草图，这样会比较形象，在计算时再注意矩阵和数组的起始下标，就不容易出错。

数组和广义表-习题

- 4、在一个二维数组A中，假设每个数组元素的长度为3个存储单元，行下标i为0~8，列下标j为0~9，从首地址SA开始连续存放。在这种情况下，元素A[8][5]的起始地址为（ ）。
A. SA+141 B. SA+144 C. SA+222 D. SA+255
- 5、设有一个12x12的对称矩阵M，将其上三角部分的元素 $m_{i,j}$ ($1 \leq i \leq j \leq 12$) 按行优先存入C语言的一维数组N中，元素 $m_{6,6}$ 在N中的下标是（ ）。
A. 50 B. 51 C. 55 D. 66
- 6、有一个100阶的三对角矩阵M，其元素 $m_{i,j}$ ($1 \leq i, j \leq 100$) 按行优先依次压缩存入下标从0开始的一维数组N中。元素 $m_{30,30}$ 在N中的下标是（ ）。
A. 86 B. 87 C. 88 D. 89

数组和广义表-习题

- 4、在一个二维数组A中，假设每个数组元素的长度为3个存储单元，行下标i为0~8，列下标j为0~9，从首地址SA开始连续存放。在这种情况下，元素A[8][5]的起始地址为（ ）。

A. SA+141 B. SA+144 C. SA+222 D. SA+255

二维数组计算地址（按行优先顺序）的公式为

$$\text{LOC}(i, j) = \text{LOC}(0, 0) + (i \times m + j) \times L$$

其中， $\text{LOC}(0, 0) = \text{SA}$ ，是数组存放的首地址； $L=3$ 是每个数组元素的长度； $m=9-0+1=10$ 是数组的列数。因此有

$$\text{LOC}(8, 5) = \text{SA} + (8 \times 10 + 5) \times 3 = \text{SA} + 255, \text{ 故选D.}$$

数组和广义表-习题

- 5、设有一个 12×12 的对称矩阵M，将其上三角部分的元素 $m_{i,j}$ ($1 \leq i \leq j \leq 12$) 按行优先存入C语言的一维数组N中，元素 $m_{6,6}$ 在N中的下标是()。

A. 50 B. 51 C. 55 D. 66

在C语言中，数组N的下标从0开始。第一个元素 $m_{1,1}$ ，对应存入n，矩阵M的第一行有12个元素，第二行有11个，第三行有10个，第四行有9个，第五行有8个，所以 $m_{6,6}$ 是第 $12+11+10+9+8+1=51$ 个元素，下标应为50。

数组和广义表-习题

- 6、有一个100阶的三对角矩阵M，其元素 $m_{i,j}$ ($1 \leq i, j \leq 100$) 按行优先依次压缩存入下标从0开始的一维数组N中。元素 $m_{30,30}$ 在N中的下标是（ ）。

A. 86

B. 87

C. 88

D. 89

$$\begin{bmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & 0 & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ & & & & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

采用压缩存储, 将3条对角线上的元素按行优先方式存放在一维数组B中, 且 $a_{1,1}$ 存放于B[0]中, 观察上图的三对角矩阵不难发现, 第一行有两个元素, 剩下的在元素 $m_{30,30}$ 所在行之前的28行 (注意下标 $1 \leq i, j \leq 100$) 中, 每行都有3个元素, 而 $m_{30,30}$ 之前仅有一个元素 $m_{30,29}$ 不难发现元素 $m_{30,30}$ 在数组N中的下标是 $2 + 28 \times 3 + 2 - 1 = 87$ 。

主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找
- 排序

树和二叉树-主要内容

- 树的定义和基本术语
- 二叉树及其存储结构
- 遍历二叉树
- 线索二叉树
- 赫夫曼树及其应用
- 树和森林
- 二叉树和树的应用示例

树和二叉树-习题

1. 度为4、高度为 h 的树，()。

A. 至少有 $h+3$ 个结点

B. 至多有 $4h-1$ 个结点

C. 至多有 $4h$ 个结点

D. 至少有 $h+4$ 个结点

2. 设一棵非空完全二叉树 T 的所有叶结点均位于同一层，且每个非叶结点都有2个子结点。若 T 有 k 个叶结点，则 T 的结点总数是()。

A. $2k-1$

B. $2k$

C. k^2

D. 2^k-1

树和二叉树-习题

1. 度为4、高度为 h 的树, (A)。

A. 至少有 $h+3$ 个结点

B. 至多有 $4h-1$ 个结点

C. 至多有 $4h$ 个结点

D. 至少有 $h+4$ 个结点

满足 1) 至少一个节点度为4 2) 每层的节点数尽可能少

2. 设一棵非空完全二叉树 T 的所有叶结点均位于同一层, 且每个非叶结点都有2个子结点。若 T 有 k 个叶结点, 则 T 的结点总数是(A)

。

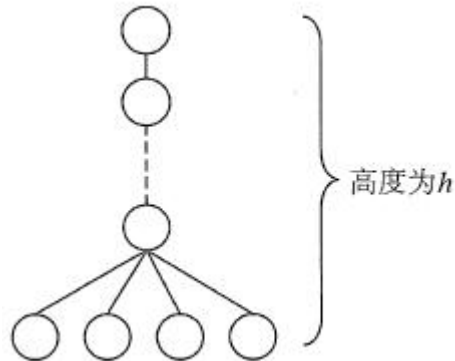
A. $2k-1$

B. $2k$

C. k^2

D. 2^k-1

$n_0-1=n_2$



树和二叉树-习题

3. 在任何一棵二叉树中，若结点a有左孩子b、右孩子c，则在结点的先序序列、中序序列、后序序列中，（ ）。

- A. 结点b一定在结点a的前面
- B. 结点a一定在结点c的前面
- C. 结点b一定在结点c的前面
- D. 结点a一定在结点b的前面

4. 若X是后序线索二叉树中的叶结点，且X存在左兄弟结点Y，则X的右线索指向的是（ ）。

- A. X的父结点
- B. 以Y为根的子树的最左下结点
- C. X的左兄弟结点Y
- D. 以Y为根的子树的最右下结点

树和二叉树-习题

3. 在任何一棵二叉树中，若结点a有左孩子b、右孩子c，则在结点的先序序列、中序序列、后序序列中，（ C ）。

A. 结点b一定在结点a的前面

B. 结点a一定在结点c的前面

C. 结点b一定在结点c的前面

D. 结点a一定在结点b的前面

都需要先遍历左子树，再遍历右子树

4. 若X是后序线索二叉树中的叶结点，且X存在左兄弟结点Y，则X的右线索指向的是（ A ）。

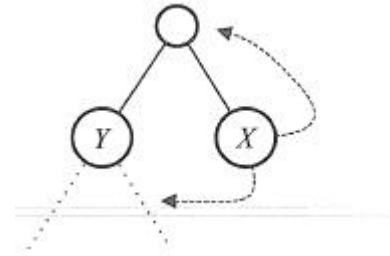
A. X的父结点

B. 以Y为根的子树的最左下结点

C. X的左兄弟结点Y

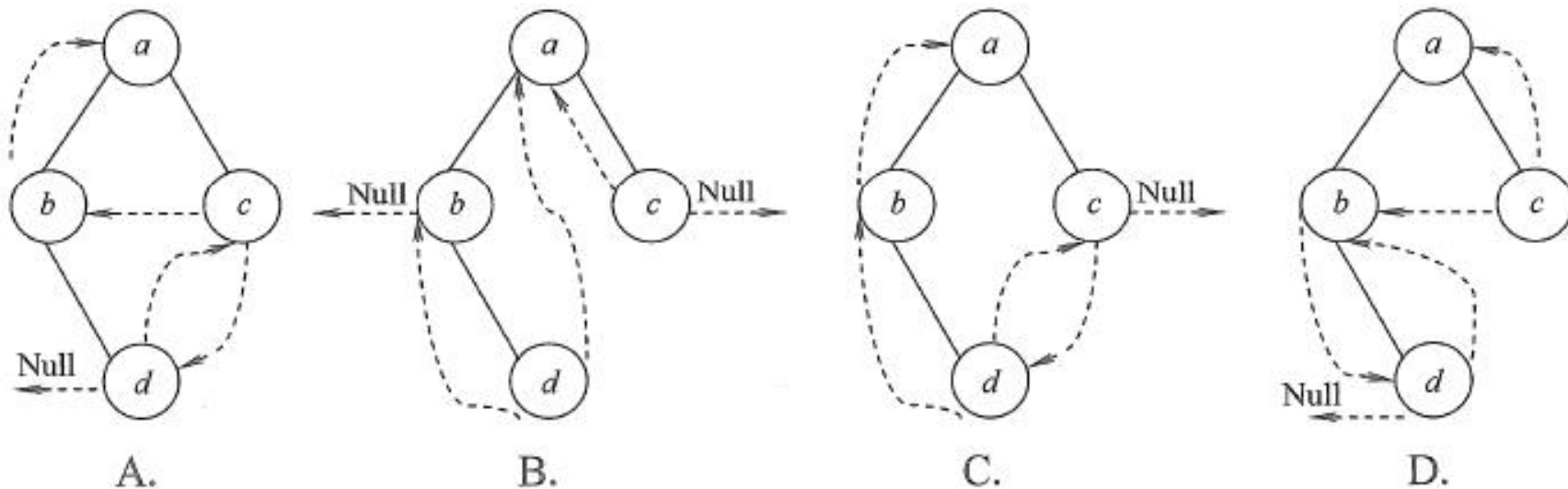
D. 以Y为根的子树的最右下结点

左线索是前驱节点，右线索后继节点。



树和二叉树-习题

5. 下列线索二叉树中(用虚线表示线索), 符合后序线索树定义的是()



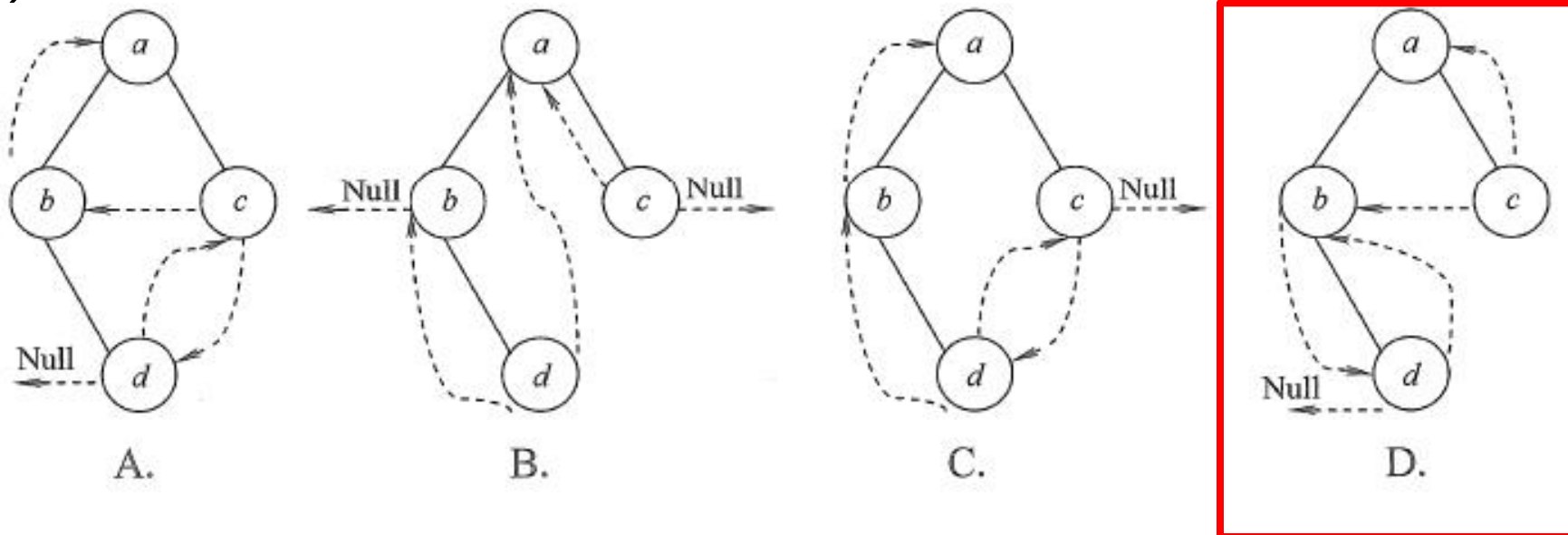
6. 要使一棵非空二叉树的先序序列与中序序列相同, 其所有非叶结点须满足的条件是()。

- A. 只有左子树
- C. 结点的度均为1

- B. 只有右子树
- D. 结点的度均为2

树和二叉树-习题

5. 下列线索二叉树中(用虚线表示线索), 符合后序线索树定义的是 (D)



dbca

6. 要使一棵非空二叉树的先序序列与中序序列相同, 其所有非叶结点须满足的条件是 (B)。

A. 只有左子树

B. 只有右子树

C. 结点的度均为1

D. 结点的度均为2

树和二叉树-习题

7. 设森林F对应的二叉树为B, 它有 m 个结点, B的根为 p , p 的右子树结点个数为 n , 森林F中第一棵树的结点个数是()。

A. $m-n$

B. $m-n-1$

C. $n+1$

D. 条件不足, 无法确定

8. 若将一棵树T转化为对应的二叉树BT, 则下列对BT的遍历中, 其遍历序列与T的后根遍历序列相同的是()。

A. 先序遍历

B. 中序遍历

C. 后序遍历

D. 按层遍历

树和二叉树-习题

7. 设森林F对应的二叉树为B, 它有 m 个结点, B的根为 p , p 的右子树结点个数为 n , 森林F中第一棵树的结点个数是(A)。

A. $m-n$

B. $m-n-1$

C. $n+1$

D. 条件不足, 无法确定

森林转化成二叉树之后, 根节点的左子树是第一棵树

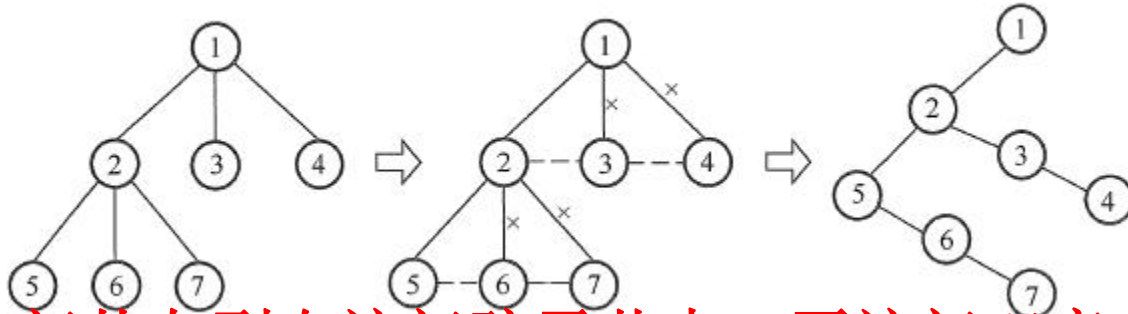
8. 若将一棵树T转化为对应的二叉树BT, 则下列对BT的遍历中, 其遍历序列与T的后根遍历序列相同的是(B)。

A. 先序遍历

B. 中序遍历

C. 后序遍历

D. 按层遍历



后根: 先访问从左到右访问孩子节点, 再访问双亲节点

树和二叉树-习题

9. 若度为 m 的哈夫曼树中，叶子结点个数为 n ，则非叶子结点的个数为()。

A. $n-1$

B. $\lfloor n/m \rfloor - 1$

C. $\lfloor (n-1)/(m-1) \rfloor$

D. $\lfloor n/(m-1) \rfloor - 1$

10. 一棵哈夫曼树共有215个结点，对其进行哈夫曼编码，共能得到()个不同的码字

A. 107

B. 108

C. 214

D. 215

树和二叉树-习题

9. 若度为 m 的哈夫曼树中，叶子结点个数为 n ，则非叶子结点的个数为(C)。

A. $n-1$

B. $\lfloor n/m \rfloor - 1$

C. $\lfloor (n-1)/(m-1) \rfloor$

D. $\lfloor n/(m-1) \rfloor - 1$

一棵度为 m 的哈夫曼树应只有度为0和 m 的结点

$$n = n_0 + n_m$$

$$mn_m = n - 1$$

10. 一棵哈夫曼树共有215个结点，对其进行哈夫曼编码，共能得到(B)个不同的码字

A. 107

B. 108

C. 214

D. 215

默认度为2

树和二叉树-习题

11. 已知字符集 {a, b, c, d, e, f}, 若各字符出现的次数分别为 6, 3, 8, 2, 10, 4, 则对应字符集中各字符的哈夫曼编码可能是().

- A. 00, 1011, 01, 1010, 11, 100
- B. 00, 100, 110, 000, 0010, 01
- C. 10, 1011, 11, 0011, 00, 010
- D. 0011, 10, 11, 0010, 01, 000

树和二叉树-习题

11. 已知字符集 {a, b, c, d, e, f}, 若各字符出现的次数分别为 6, 3, 8, 2, 10, 4, 则对应字符集中各字符的哈夫曼编码可能是(A).

- A. 00, 1011, 01, 1010, 11, 100
- B. 00, 100, 110, 000, 0010, 01
- C. 10, 1011, 11, 0011, 00, 010
- D. 0011, 10, 11, 0010, 01, 000

出现的次数越多的的哈夫曼编码码长越短

树和二叉树-习题

试设计判断两棵二叉树是否相似的算法。所谓二叉树 T_1 和 T_2 相似，指的是 T_1 和 T_2 都是空的二叉树或都只有一个根结点；或 T_1 的左子树和 T_2 的左子树是相似的且 T_1 的右子树和 T_2 的右子树是相似的。

树和二叉树-习题

试设计判断两棵二叉树是否相似的算法。所谓二叉树T1和T2相似，指的是T1和T2都是空的二叉树或都只有一个根结点；或T1的左子树和T2的左子树是相似的且T1的右子树和T2的右子树是相似的

本题采用递归的思想求解，若T1和T2都是空树，则相似；若有一个为空另一个不空，则必然不相似；否则递归地比较它们的左、右子树是否相似。递归函数的定义如下：

- a) $f(T1, T2) = 1$ ；若 $T1 == T2 == \text{NULL}$.
- b) $f(T1, T2) = 0$ ；若T1和T2之一为NULL, 另一个不为NULL.
- c) $f(T1, T2) = f(T1 \rightarrow \text{lchild}, T2 \rightarrow \text{lchild}) \&\& (T1 \rightarrow \text{rchild}, T2 \rightarrow \text{rchild})$ ；若T1和T2均不为NULL.

主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找
- 排序

图-主要内容

- 图的各种存储结构及其构造算法，实际问题的求解效率与采用何种存储结构和算法有密切联系。
- 遍历的逻辑定义、深度优先搜索和广度优先搜索的算法。
- 图的遍历算法与树的遍历算法之间的类似和差异。
- 最小连通图、拓扑排序、关键路径、最短路径等图应用问题的算法原理和执行过程。

图-习题

- 1. 一个有 n 个顶点和 n 条边的无向图一定是()。
 - A. 连通的 B. 不连通的 C. 无环的 D. 有环的
- 2. 若从无向图的任意顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是()。
 - A. 强连通图 B. 连通图 C. 有回路 D. 一棵树
- 3. 以下关于图的叙述中，正确的是()。
 - A. 图与树的区别在于图的边数大于等于顶点数.
 - B. 假设有图 $G = \{V, \{E\}\}$ ，顶点集 $V' \in V$ ， $E' \in E$ ，则 V 和 $\{E'\}$ 构成 G 的子图
 - C. 无向图的连通分量是指无向图中的极大连通子图
 - D. 图的遍历就是从图中某一顶点出发访遍图中其余顶点

图-习题

■ 1. 一个有 n 个顶点和 n 条边的无向图一定是()。

A. 连通的 B. 不连通的 C. 无环的 D. 有环的

若一个无向图有 n 个顶点和 $n-1$ 条边，可以使它连通但没有环(即生成树)，但再加一条边，在不考虑重边的情形下，则必然会构成环。

■ 2. 若从无向图的任意顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是()。

A. 强连通图 B. 连通图 C. 有回路 D. 一棵树

强连通图是有向图，与题意矛盾，选项A错误；对无向连通图做一次深度优先搜索，可以访问到该连通图的所有顶点，选项B正确；有回路的无向图不一定是连通图，因为回路不一定包含图的所有结点，选项C错误；连通图可能是树，也可能存在环，选项D错误。

图-习题

■ 3. 以下关于图的叙述中，正确的是()。

- A. 图与树的区别在于图的边数大于等于顶点数.
- B. 假设有图 $G = \{V, \{E\}\}$ ，顶点集 $V' \in V$ ， $E' \in E$ ，则 V 和 $\{E'\}$ 构成 G 的子图
- C. 无向图的连通分量是指无向图中的极大连通子图
- D. 图的遍历就是从图中某一顶点出发访遍图中其余顶点

图与树的区别是逻辑上的区别，而不是边数的区别，图的边数也可能小于树的边数，故选项A错；若 E' 中的边对应的顶点不是 V' 的元素， V' 和 $\{E'\}$ 无法构成图，故选项B错；无向图的极大连通子图称为连通分量，选项C正确；图的遍历要求每个结点只能被访问一次，且若图非连通，则从某一顶点出发无法访问到其他全部顶点，选项D的说法不准确。

图-习题

- 4. 下列关于图的叙述中，正确的是()。
- I. 回路是简单路径
 - II. 存储稀疏图，用邻接矩阵比邻接表更省空间
 - III. 若有向图中存在拓扑序列，则该图不存在回路
- A. 仅II B. 仅I、II C. 仅III D. 仅I、III
- 5. 下列关于无向连通图特性的叙述中，正确的是()
- I. 所有顶点的度之和为偶数
 - II. 边数大于顶点个数减1
 - III. 至少有一个顶点的度为1
- A. 只有I B. 只有II C. I和II D. I和III

图-习题

■ 4. 下列关于图的叙述中，正确的是()。

I. 回路是简单路径

II. 存储稀疏图，用邻接矩阵比邻接表更省空间

III. 若有向图中存在拓扑序列，则该图不存在回路

A. 仅II B. 仅I、II C. 仅III D. 仅I、III

回路对应于路径，简单回路对应于简单路径，故 I 错误；稀疏图是边比较少的情況，此时用邻接矩阵必将浪费大量的空间，应选用邻接表，故 II 错误。存在回路的图不存在拓扑序列，III 正确。

图-习题

- 5. 下列关于无向连通图特性的叙述中，正确的是()
- I. 所有顶点的度之和为偶数
 - II. 边数大于顶点个数减1
 - III. 至少有一个顶点的度为1
- A. 只有I B. 只有II C. I和II D. I和III

无向连通图对应的生成树也是无向连通图，但此时边数等于顶点数减1，故I错误。考虑一个无向连通图的顶点恰好构成一个回路的情况，此时每个顶点的度都是2，故III错误。在无向图中，所有顶点的度之和为边数的2倍，故 I 正确。

图-习题

- 6. 若无向图 $G=(V, E)$ 中含有7个顶点，要保证图 G 在任何情况下都是连通的，则需要的边数最少是()。
A. 6 B. 15 C. 16 D. 21
- 7. 以下关于图的叙述中，正确的是()。
A. 强连通有向图的任何顶点到其他所有顶点都有弧
B. 图的任意顶点的入度等于出度
C. 有向完全图一定是强连通有向图
D. 有向图的边集的子集和顶点集的子集可构成原有有向图的子图
- 8. 一个有28条边的非连通无向图至少有()个顶点。
A. 7 B. 8 C. 9 D. 10

图-习题

- 6. 若无向图 $G=(V, E)$ 中含有7个顶点，要保证图 G 在任何情况下都是连通的，则需要的边数最少是()。

A. 6 B. 15 **C. 16** D. 21

题干要求在“任何情况”下都是连通的，考虑最极端的情形，即图 G 的6个顶点构成一个完全无向图，再加上一条边后，第7个顶点必然与此完全无向图构成一个连通图，所以最少边数 $=6 \times 5 / 2 + 1 = 16$ 。若边数 n 小于等于15，可以使这 n 条边仅连接图 G 中的某6个顶点，从而导致第7个顶点无法与这6个顶点构成连通图（不满足“任何情况”）。

图-习题

■ 7. 以下关于图的叙述中，正确的是()。

- A. 强连通有向图的任何顶点到其他所有顶点都有弧
- B. 图的任意顶点的入度等于出度
- C. 有向完全图一定是强连通有向图
- D. 有向图的边集的子集和顶点集的子集可构成原有有向图的子图

强连通有向图的任何顶点到其他所有顶点都有路径，但未必有弧；无向图任意顶点的入度等于出度，但有向图未必满足；若边集中的某条边对应的某个顶点不在对应的顶点集中，则有向图的边集的子集和顶点集的子集无法构成子图。

图-习题

- 8. 一个有28条边的非连通无向图至少有()个顶点。
- A. 7 B. 8 **C. 9** D. 10

考虑该非连通图最极端的情况，即它由一个完全图加一个独立的顶点构成，此时若再加一条边，则必然使图变成连通图。在 $28 = n(n-1)/2 = 8 \times 7/2$ 条边的完全无向图中，总共有8个顶点，再加上1个不连通的顶点，共9个顶点。

图-习题

- 9. 在有 n 个顶点的有向图中，每个顶点的度最大可达()。
A. n B. $n-1$ C. $2n$ D. $2n-2$
- 10. 具有6个顶点的无向图，当有()条边时能确保是一个连通图。
A. 8 B. 9 C. 10 D. 11
- 11. 已知无向图 G 含有16条边，其中度为4的顶点个数为3, 度为3的顶点个数为4，其他顶点的度均小于3。图 G 所含的顶点个数至少是()。
A. 10 B. 11 C. 13 D. 15

图-习题

■ 9. 在有 n 个顶点的有向图中，每个顶点的度最大可达()。

- A. n B. $n-1$ C. $2n$ D. $2n-2$

在有向图中，顶点的度等于入度与出度之和。 n 个顶点的有向图中，任意一个顶点最多还可以与其他 $n-1$ 个顶点有一对指向相反的边相连。

■ 10. 具有6个顶点的无向图，当有()条边时能确保是一个连通图。

- A. 8 B. 9 C. 10 D. 11

解题思路与第7题类似。5个顶点构成一个完全无向图，需要10条边；再加上1条边后，能保证第6个顶点必然与此完全无向图构成一个连通图，故共需11条边。

图-习题

- 11. 已知无向图G含有16条边，其中度为4的顶点个数为3，度为3的顶点个数为4，其他顶点的度均小于3。图G所含的顶点个数至少是()。

A. 10 B. 11 C. 13 D. 15

无向图边数的2倍等于各顶点度数的总和。由于其他顶点的度均小于3，设它们的度都为2，设它们的数量是x，列出这方程 $4 \times 3 + 3 \times 4 + 2x = 16 \times 2$ ，解得 $x=4$ 。 $4+4+3=11$ ，选项B正确。

图-习题

■ 12. 在含有 n 个顶点和 e 条边的无向图的邻接矩阵中，零元素的个数为()。

A. e B. $2e$ C. $n^2 - e$ D. $n^2 - 2e$

■ 13. 带权有向图 G 用邻接矩阵存储，则 v_i 的入度等于邻接矩阵中()。

A. 第 i 行非无穷的元素个数

B. 第 i 列非无穷的元素个数

C. 第 i 行非无穷且非0的元素个数

D. 第 i 列非无穷且非0的元素个数

图-习题

■ 12. 在含有 n 个顶点和 e 条边的无向图的邻接矩阵中，零元素的个数为()。

- A. e B. $2e$ C. n^2-e D. n^2-2e

无向图的邻接矩阵中，矩阵大小为 n^2 ，非零元素的个数为 $2e$ ，故零元素的个数为 n^2-2e 。读者应掌握此题的变体，即当无向图变为有向图时，能够求出零的个数和非零的个数。

■ 13. 带权有向图 G 用邻接矩阵存储，则 v_i 的入度等于邻接矩阵中()。

- A. 第 i 行非无穷的元素个数
B. 第 i 列非无穷的元素个数
C. 第 i 行非无穷且非0的元素个数
D. 第 i 列非无穷且非0的元素个数

有向图的邻接矩阵中，0和无穷表示的都不是有向边，而入度是由邻接矩阵的列中元素计算出来的；出度是由邻接矩阵的行中元素计算出来的。

图-习题

■ 14. 下列关于广度优先算法的说法中, 正确的是()。

I. 当各边的权值相等时, 广度优先算法可以解决单源最短路径问题

II. 当各边的权值不等时, 广度优先算法可用来解决单源最短路径问题

III. 广度优先遍历算法类似于树中的后序遍历算法

IV. 实现图的广度优先算法时, 使用的数据结构是队列

A. I、IV B. I、II、IV C. II、IV D. I、III、IV

■ 15. 对一个有 n 个顶点、 e 条边的图采用邻接表表示时, 进行DFS遍历的时间复杂度为()空间复杂度为();进行BFS遍历的时间复杂度为(), 空间复杂度为()。

A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(1)$

图-习题

■ 14. 下列关于广度优先算法的说法中, 正确的是()。

I. 当各边的权值相等时, 广度优先算法可以解决单源最短路径问题

II. 当各边的权值不等时, 广度优先算法可用来解决单源最短路径问题

III. 广度优先遍历算法类似于树中的后序遍历算法

IV. 实现图的广度优先算法时, 使用的数据结构是队列

A. I、IV B. I、II、IV C. II、IV D. I、III、IV

广度优先搜索以起始结点为中心, 一层一层地向外层扩展遍历图的顶点, 因此无法考虑到边权值, 只适合求边权值相等的图的单源最短路径。广度优先搜索相当于树的层序遍历, III错误。广度优先搜索需要用到队列, 深度优先搜索需要用到栈, I、IV正确。

图-习题

■ 15. 对一个有 n 个顶点、 e 条边的图采用邻接表表示时，进行DFS遍历的时间复杂度为()空间复杂度为()；进行BFS遍历的时间复杂度为()，空间复杂度为()。

A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(1)$

C、A、C、A

深度优先遍历时，每个顶点表结点和每个边表结点均查找一次，每个顶点递归调用一次，需要借助一个递归工作栈；而广度优先遍历时，也是每个顶点表结点和每个边表结点均查找一次，每个顶点进入队列一次。故都是选C, A。

图-习题

■ 16. 对有 n 个顶点、 e 条边的图采用邻接矩阵表示时，进行DFS遍历的时间复杂度为()，进行BFS遍历的时间复杂度为()。

A. $O(n)$ B. $O(e)$ C. $O(n+e)$ D. $O(e^2)$

■ 17. 下列选项中，不是下图深度优先搜索序列的是（）。

A. V_1, V_5, V_4, V_3, V_2 B. V_1, V_3, V_2, V_5, V_4

C. V_1, V_2, V_5, V_4, V_3 D. V_1, V_2, V_3, V_4, V_5

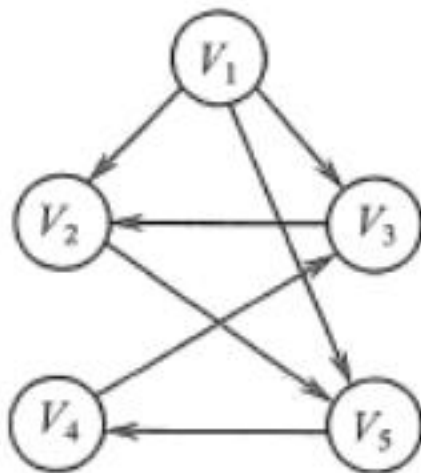


图-习题

■ 16. 对有 n 个顶点、 e 条边的图采用邻接矩阵表示时，进行DFS遍历的时间复杂度为()，进行BFS遍历的时间复杂度为()。

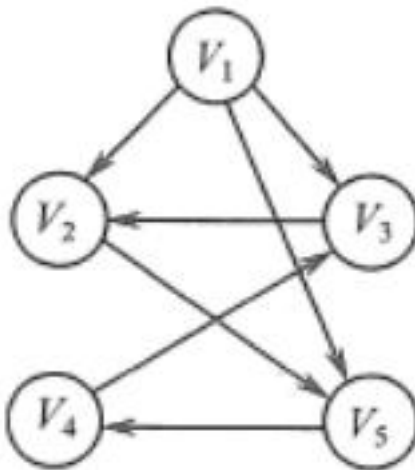
A. $O(n^2)$ B. $O(e)$ C. $O(n+e)$ D. $O(e^2)$

A、A

采用邻接矩阵表示时，查找一个顶点所有出边的时间复杂度为 $O(n)$ ，共有 n 个顶点，故时间复杂度均为 $O(n^2)$ 。

图-习题

- 17. 下列选项中，不是下图深度优先搜索序列的是（ ）。



- A. V_1, V_5, V_4, V_3, V_2 B. V_1, V_3, V_2, V_5, V_4
C. V_1, V_2, V_5, V_4, V_3 **D. V_1, V_2, V_3, V_4, V_5**

图-习题

■ 18. 设有向图 $G=(V, E)$, 顶点集 $V=\{V_0, V_1, V_2, V_3\}$, 边集 $E=\{\langle V_0, V_1 \rangle, \langle V_0, V_2 \rangle, \langle V_0, V_3 \rangle, \langle V_1, V_3 \rangle\}$ 。若从顶点 V_0 开始对图进行深度优先遍历, 则可能得到的不同遍历序列个数是()。

- A. 2 B. 3 C. 4 D. 5

图-习题

■ 18. 设有向图 $G=(V, E)$, 顶点集 $V=\{V_0, V_1, V_2, V_3\}$, 边集 $E=\{\langle V_0, V_1 \rangle, \langle V_0, V_2 \rangle, \langle V_0, V_3 \rangle, \langle V_1, V_3 \rangle\}$ 。若从顶点 V_0 开始对图进行深度优先遍历, 则可能得到的不同遍历序列个数是()。

- A. 2 B. 3 C. 4 **D. 5**

画出该有向图的图形, 如右图所示。采用图的深度优先遍历, 共有 5 种可能:
 $\langle V_0, V_1, V_3, V_2 \rangle, \langle V_0, V_2, V_3, V_1 \rangle, \langle V_0, V_2, V_1, V_3 \rangle, \langle V_0, V_3, V_2, V_1 \rangle, \langle V_0, V_3, V_1, V_2 \rangle$, 选 D。

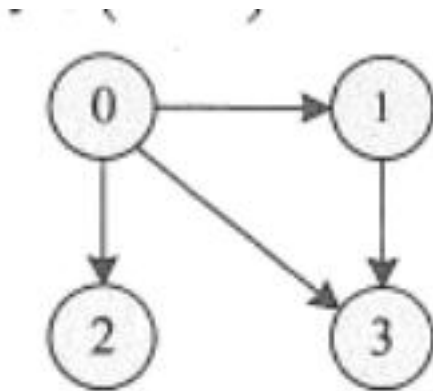


图-习题

- 19. 已知含有5个顶点的图G如下图所示。
- 请回答下列问题：
- 1) 写出图G的邻接矩阵A(行、列下标从0开始)。
- 2) 求A，矩阵A中位于0行3列元素值的含义是什么？

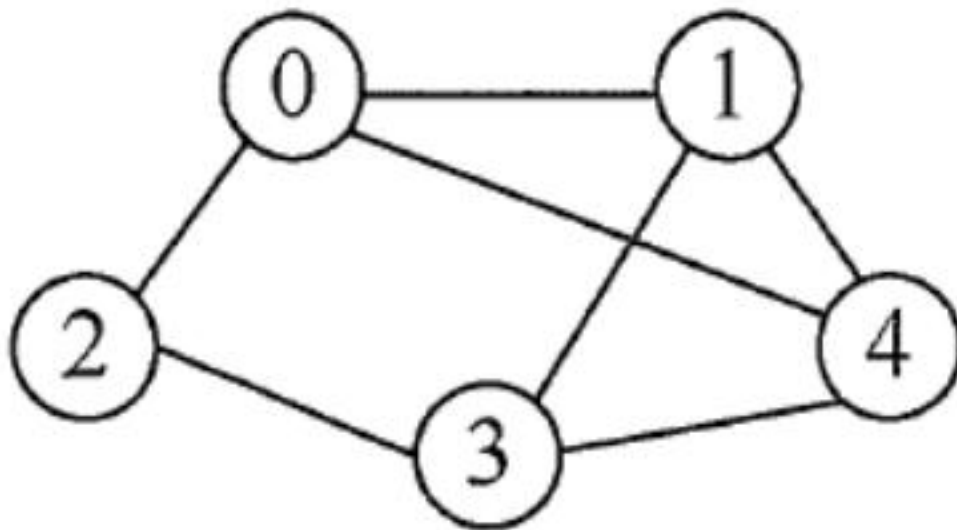


图-习题

1) 图 G 的邻接矩阵 A 如下:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

2) A^2 如下:

$$A^2 = \begin{bmatrix} 3 & 1 & 0 & 3 & 1 \\ 1 & 3 & 2 & 1 & 2 \\ 0 & 2 & 2 & 0 & 2 \\ 3 & 1 & 0 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 \end{bmatrix}$$

0 行 3 列的元素值 3 表示从顶点 0 到顶点 3 之间长度为 2 的路径共有 3 条。

主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找
- 排序

查找-主要内容

- 顺序表和有序表的查找方法及其平均查找长度的计算方法。
- 了解静态查找树的构造方法和查找方法及其和有序表的差别。

查找-主要内容

- 熟练掌握二叉排序树以及平衡二叉树的构造和查找方法。
- B-树和B+树的结构特点以及它们各自查找过程的不同之处。
- 哈希表的构造方法，深刻理解哈希表与其它结构的表的实质性的差别。
- 计算各种查找方法在等概率情况下查找成功时的平均查找长度和查找失败时的平均查找长度。

查找-习题

- 1. 对长度为3的顺序表进行查找，若查找第一个元素的概率为 $1/2$ ，查找第二个元素的概率为 $1/3$ ，查找第三个元素的概率为 $1/6$ ，则查找任一元素的平均查找长度为()。
A. $5/3$ B. 2 C. $7/3$ D. $4/3$
- 2. 折半查找过程所对应的判定树是一棵 ()。
A. 最小生成树 B. 平衡二叉树
C. 完全二叉树 D. 满二叉树
- 3. 已知一个长度为16的顺序表工，其元素按关键字有序排列，若采用折半查找法查找一个工中不存在的元素，则关键字的比较次数最多是()。
A. 4 B. 5 C. 6 D. 7

查找-习题

- 1. 对长度为3的顺序表进行查找，若查找第一个元素的概率为 $1/2$ ，查找第二个元素的概率为 $1/3$ ，查找第三个元素的概率为 $1/6$ ，则查找任一元素的平均查找长度为()。

A. $5/3$ B. 2 C. $7/3$ D. $4/3$

在长度为3的顺序表中，查找第一个元素的查找长度为1，查找第二个元素的查找长度为2，查找第三个元素的查找长度为3，故有

$$ASL_{\text{成功}} = 1/2 * 1 + 1/3 * 2 + 1/6 * 3 = 5/3$$

- 2. 折半查找过程所对应的判定树是一棵 ()。

A. 最小生成树 B. 平衡二叉树
C. 完全二叉树 D. 满二叉树

查找-习题

由折半查找的定义不难看出，每次把一个数组从中间结点分割时，总是把数组分为结点数相差最多不超过1的两个子数组，从而使得对应的判定树的两棵子树高度差的绝对值不超过1，所以应是平衡二叉树。

- 3. 已知一个长度为16的顺序表工，其元素按关键字有序排列，若采用折半查找法查找一个工中不存在的元素，则关键字的比较次数最多是()。

A. 4 B. 5 C. 6 D. 7

折半查找法在查找不成功时和给定值进行关键字的比较次数最多为树的高度，即 $\lceil \log_2 n \rceil + 1$ 或 $\lceil \log_2 (n + 1) \rceil$ 。在本题中， $n = 16$ ，故比较次数最多为5。

查找-习题

■ 4. 下列选项中，不能构成折半查找中关键字比较序列的是。

A. 500, 200, 450, 180

B. 500, 450, 200, 180

C. 180, 500, 200, 450

D. 180, 200, 500, 450

■ 5. 在有 n ($n > 1000$) 个元素的升序数组 A 中查找关键字 X 。查找算法的伪代码如下所示。

```
k=0;    while(k<n && a[k]<x) k=k+3;
if(k<n && a[k]==x) 查找成功;
else if(k-1<n && a[k-1]==x) 查找成功;
    else if(k-2<n && a[k-2]==x) 查找成功;
        else 查找失败;
```

本算法与折半查找算法相比，有可能更少比较次数的情况是

A. 当 x 不在数组中 B. 当 x 接近数组开头处

C. 当 x 接近数组结尾处 D. 当 x 处于数组中间位置

查找-习题

■ 4. 下列选项中，不能构成折半查找中关键字比较序列的是。

A. 500, 200, 450, 180

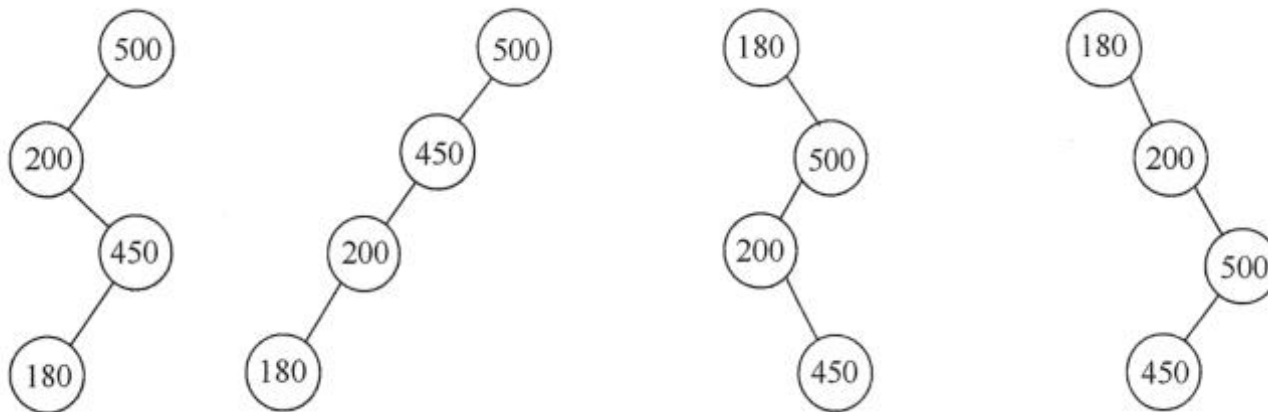
B. 500, 450, 200, 180

C. 180, 500, 200, 450

D. 180, 200, 500, 450

如下图所示，画出查找路径图，因为折半查找的判定树是一棵二叉排序树

■ 5. B



查找-习题

- 6. 下列叙述中，不符合 m 阶B树定义要求的是()。
 - A. 根结点至多有 m 棵子树
 - B. 所有叶结点都在同一层上
 - C. 各结点内关键字均升序或降序排列
 - D. 叶结点之间通过指针链接
- 7. 在一棵具有15个关键字的4阶B树中，含关键字的结点个数最多是()。
 - A. 5
 - B. 6
 - C. 10
 - D. 14

查找-习题

- 6. 下列叙述中，不符合 m 阶B树定义要求的是()。
 - A. 根结点至多有 m 棵子树
 - B. 所有叶结点都在同一层上
 - C. 各结点内关键字均升序或降序排列
 - D. 叶结点之间通过指针链接(D描述的是B+树)
- 7. 在一棵具有15个关键字的4阶B树中，含关键字的结点个数最多是()。
 - A. 5
 - B. 6
 - C. 10
 - D. 14

对于5阶B树，根结点只有达到5个关键字时才能产生分裂，成为高度为2的B树，因此高度为2的5阶B树所含关键字的个数最少是5。

查找-习题

- 8. 为提高散列表的查找效率，可以采取的正确措施是()。
- I. 增大装填（载）因子
 - II. 设计冲突（碰撞）少的散列函数
 - III. 处理冲突（碰撞）时避免产生聚集（堆积）现象
- A. 仅I B. 仅II C. 仅I、II D. 仅II、III
- 9. 现有长度为11且初始为空的散列表HT，散列函数是 $H(\text{key}) = \text{key} \% 7$ ，采用线性探查（线性探测再散列）法解决冲突。将关键字序列87, 40, 30, 6, 11, 22, 98, 20依次插入HT后，HT查找失败的平均查找长度是()。
- A. 4 B. 5.25 C. 6 D. 6.29

查找-习题

■ 8. 为提高散列表的查找效率，可以采取的正确措施是()。

I. 增大装填（载）因子

II. 设计冲突（碰撞）少的散列函数

III. 处理冲突（碰撞）时避免产生聚集（堆积）现象

A. 仅I

B. 仅 II

C. 仅I、 II

D. 仅 II、 III

散列表的查找效率取决于：散列函数、处理冲突的方法和装填因子。显然，冲突的产生概率与装填因子（即表中记录数与表长之比）的大小成正比，I 与题意相反。I 显然正确。采用合适的冲突处理方法可避免聚集现象，也将提高查找效率，I 正确。例如，用链地址法处理冲突时不存在聚集现象，用线性探测法处理冲突时易引起聚集现象。

查找-习题

- 9. 现有长度为11且初始为空的散列表HT, 散列函数是 $H(key)=key\%7$, 采用线性探查（线性探测再散列）法解决冲突。将关键字序列87, 40, 30, 6, 11, 22, 98, 20依次插入HT后, HT查找失败的平均查找长度是()。

A. 4

B. 5.25

C. 6 D. 6.29

采用线性探查法计算每个关键字的存放情况如下表所示。由于 $H(key)=0\sim 6$, 查找失败时可能对应的地址有7个, 对于计算出地址为0的关键字key0, 只有比较完0~8号地址后才能确定该关键字不在表中, 比较次数为9; 对于计算出地址为1的关键字key1, 只有比较完1~8号地址后才能确定该关键字不在表中, 比较次数为8; 以此类推。需要特别注意的是, 散列函数不可能计算出地址7, 因此有

$$ASL_{\text{失败}} = (9+8+7+6+5+4+3)/7=6$$

散列地址	0	1	2	3	4	5	6	7	8	9	10
关键字	98	22	30	87	11	40	6	20			

查找-习题

- 10. 设包含4个数据元素的集合 $S = \{ 'do', 'for', 'repeat', 'while' \}$ ，各元素的查找概率依次为 $p_1=0.35, p_2=0.15, p_3=0.15, p_4=0.35$ 。将 S 保存在一个长度为4的顺序表中，采用折半查找法，查找成功时的平均查找长度为2.2。
- 1) 若采用顺序存储结构保存 S ，且要求平均查找长度更短，则元素应如何排列？应使用何种查找方法？查找成功时的平均查找长度是多少？
 - 2) 若采用链式存储结构保存 S ，且要求平均查找长度更短，则元素应如何排列？应使用何种查找方法？查找成功时的平均查找长度是多少？

查找-习题

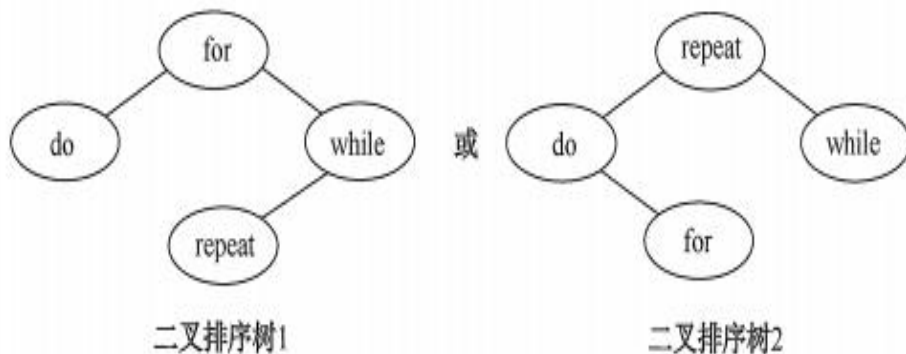
10. 答案

1) 折半查找要求元素有序顺序存储, 若各个元素的查找概率不同, 折半查找的性能不一定优于顺序查找。采用顺序查找时, 元素按其查找概率的降序排列时查找长度最小。采用顺序存储结构, 数据元素按其查找概率降序排列。采用顺序查找方法。查找成功时的平均查找长度 $=0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。此时, 显然查找长度比折半查找的更短。

2) 答案一: 采用链式存储结构时, 只能采用顺序查找, 其性能和顺序表一样, 类似于上题。数据元素按其查找概率降序排列, 构成单链表。采用顺序查找方法。查找成功时的平均查找长度 $=0.35 \times 1 + 0.35 \times 2 + 0.15 \times 3 + 0.15 \times 4 = 2.1$ 。

答案二: 还可以构造成二叉排序树的形式。采用二叉链表的存储结构, 构造二叉排序树, 元素的存储方式见下图。采用二叉排序树的查找方法。

查找成功时的平均查找长度 $=0.15 \times 1 + 0.35 \times 2 + 0.35 \times 2 + 0.15 \times 3 = 2.0$ 。



主要内容

- 线性表
- 栈和队列
- 串
- 数组和广义表
- 树和二叉树
- 图
- 查找表
- 排序

排序-习题

对一组数据（2，12，16，88，5，10）进行排序，若前3趟排序结果如下：

第一趟排序结果：2，12，16，5，10，88

第二趟排序结果：2，12，5，10，16，88

第三趟排序结果：2，5，10，12，16，88

则采用的排序方法可能是（ ）

A. 冒泡排序

B. 希尔排序

C. 归并排序

D. 基数排序

排序-习题

对一组数据（2，12，16，88，5，10）进行排序，若前3趟排序结果如下：

第一趟排序结果：2，12，16，5，10，88

第二趟排序结果：2，12，5，10，16，88

第三趟排序结果：2，5，10，12，16，88

则采用的排序方法可能是(A)

A. 冒泡排序

B. 希尔排序

C. 归并排序

D. 基数排序

分别用其他3种排序算法执行数据，归并排序第一趟结果为（2，12，16，88，5，10），基数排序第一趟后结果为（10，2，12，5，16，88），希尔排序显然是不符合的。只有冒泡排序符合条件。

排序-习题

在内部排序时，若选择了归并排序而没有选择插入排序，则可能的理由是()。

I、归并排序的程序代码更短。 II归并排序的占用空间更少
III. 归并排序的运行效率更高

A. 仅II B. 仅III C. 仅I、II D. 仅I、III

排序-习题

在内部排序时，若选择了归并排序而没有选择插入排序，则可能的理由是(B)。

I. 归并排序的程序代码更短. II. 归并排序的占用空间更少
III. 归并排序的运行效率更高

A. 仅II B. 仅III C. 仅I、II D. 仅I、III

归并排序代码比选择插入排序更复杂，前者空间复杂度是 $O(n)$ ，后者是 $O(1)$ 。但是前者时间复杂度是 $O(n\log n)$ ，后者是 $O(n^2)$ 。所以选项B正确。

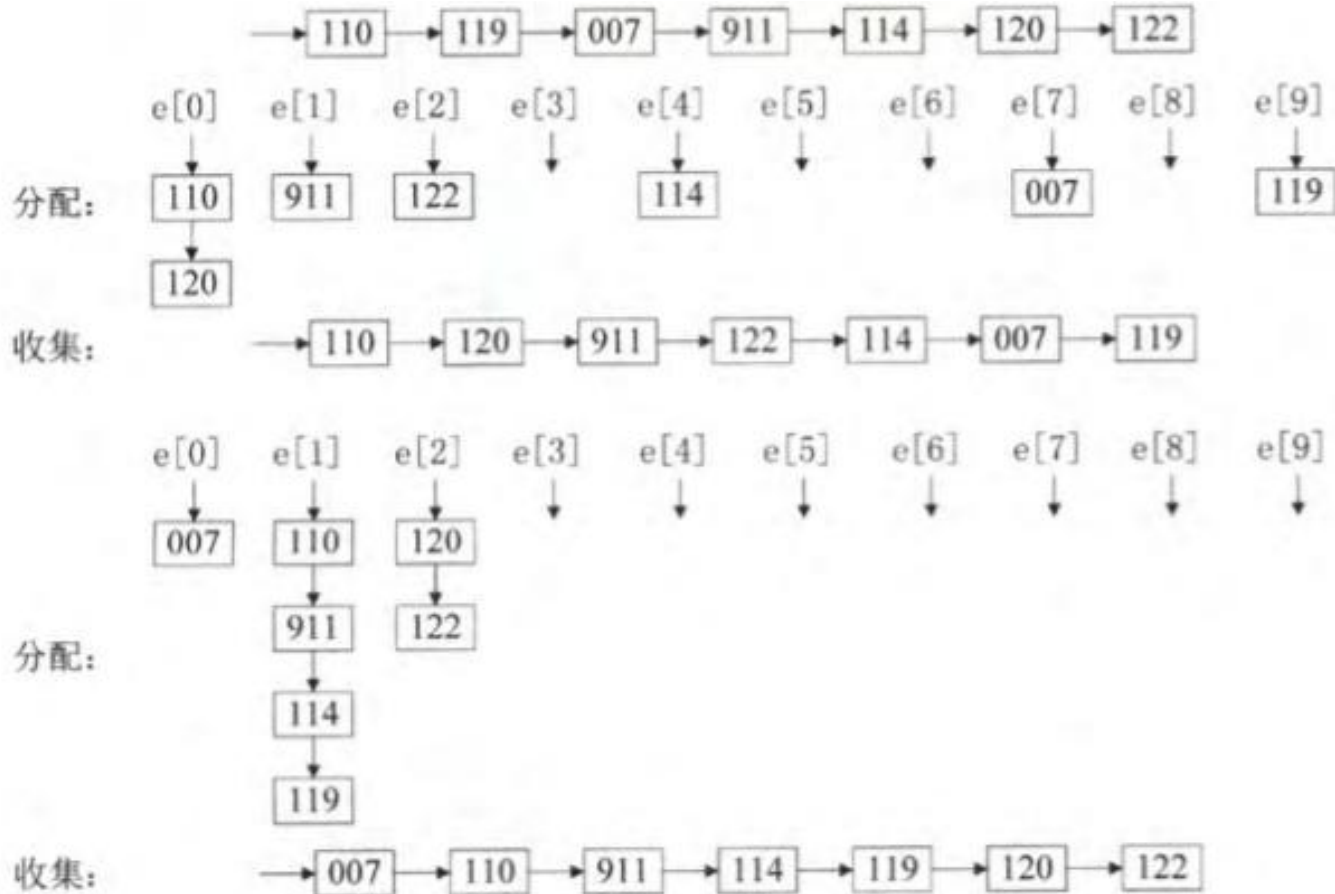
排序-习题

对给定的关键字序列110, 119, 007, 911, 114, 120, 122进行基数排序，则第2趟分配收集后得到的关键字序列是()。

- A. 007, 110, 119, 114, 911, 120, 122
- B. 007, 110, 119, 114, 911, 122, 120
- C. 007, 110, 911, 114, 119, 120, 122
- D. 110, 120, 911, 122, 114, 007, 119

排序-习题

基数排序的第1趟排序是按照个位数字的大小来排序的，第2趟排序是按照十位数字的大小进行排序的，排序的过程如下图所示。



排序-习题

下列排序算法中，元素的移动次数与关键字的初始排列次序无关的是()。

- A. 直接插入排序 B. 起泡排序 C. 基数排序 D. 快速排序

排序-习题

下列排序算法中，元素的移动次数与关键字的初始排列次序无关的是(C)。

A. 直接插入排序 B. 冒泡排序 C. 基数排序 D. 快速排序

基数排序的元素移动次数与关键字的初始排列次序无关，而其他三种排序都是与关键字的初始排列明显相关的。

排序-习题

已知三叉树T中6个叶结点的权分别是2, 3, 4, 5, 6, 7, T的带权(外部路径长度)最小是()

A. 27

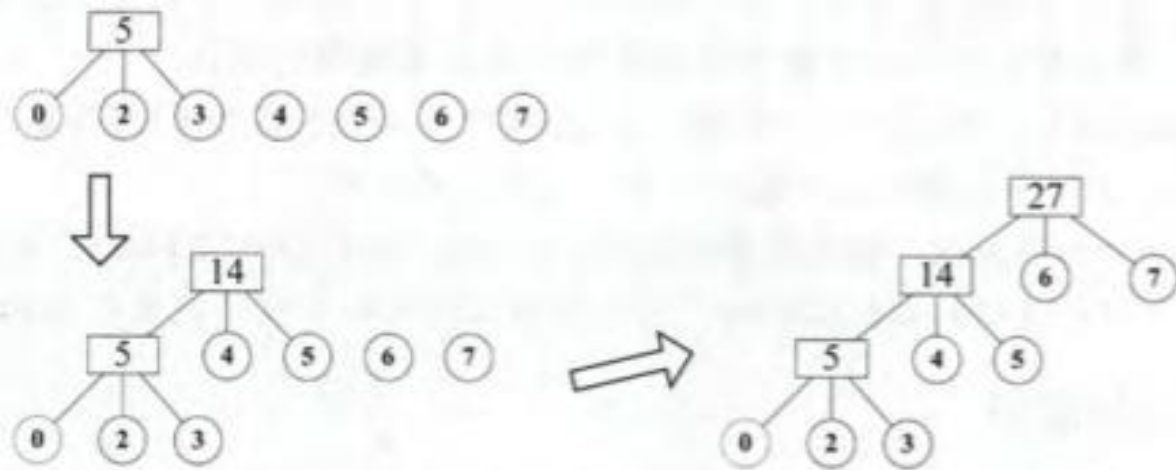
B. 46

C. 54

D. 56

排序-习题

B 将哈夫曼树的思想推广到三叉树的情形。为了构成严格的三叉树，需添加权为0的虚叶结点，对于严格的三叉树 $(n-1) \% (3-1) = u = 1$ ，需要添加 $m-u-1 = 3-1-1 = 1$ 个叶结点，说明7个叶结点刚好可以构成一个严格的三叉树。按照哈夫曼树的原则，权为0的叶结点应离树根最远，构造最小带权生成树的过程如下：



最小的带权路径长度为 $(2+3) \times 3 + (4+5) \times 2 + (6+7) \times 1 = 46$ 。

排序-习题

置换-选择排序的作用是()。

- A. 置换-选择排序用于生成外排序的初始归并段
- B. 置换-选择排序是完成将一个磁盘文件排序成有序文件的有效的外排序算法
- C. 置换-选择排序生成的初始归并段的长度是内存工作区的2倍
- D. 置换-选择排序是对外排序中输入/归并/输出的并行处理

A