**第二章作业：**

1、设计一算法，对单链表实现就地逆置。线性表（a1,a2,..an）逆置为（an,an-1,..a1）（就地含义辅助空间O(1)）

typedef struct LNode

{ ElemType data; //数据域

struct LNode \*next; //指针域

}LNode,\*LinkList

void inverse(LinkList &L)

{// 逆置带头结点的单链表 L

p=L->next; L->next=NULL;

while ( p) {

q=p->next; // q指向\*p的后继

p->next=L->next;

L->next=p; // \*p插入在头结点之后

p = q;

}

}

2、 假设以两个元素依值递增有序排列的线性表A 和B分别表示两个集合（即同一表中的元素值各不相同），

现要求另辟空间构成一个线性表C，其元素为A和B中元素的交集，且表C中的元素也依值递增有序排列。

（1）顺序表编写求C的算法。

（2）单链表编写求C的算法。

3、数组方式实现有序的顺序表合并P26 算法2.7（算法2.7是用指针方式实现）

**第三-五章作业：**

1. 有5个元素，其进栈次序为A、B、C、D、E,在各种可能的出栈次序中，以元素C、D最先出栈（即C第一个且D第二个出栈）的次序有哪几个？

．CDBAE  CDEBA CDBEA

2、使用顺序栈写出判别字符串是否回文（正写和倒写都一样例如:ABA或ABBA）的算法的程序实现。

3、使用循环队列，利用队列的基本运算返回指定队列中队尾元素。

4、写出串匹配的BF算法的完整程序，并举例画图指示解释当匹配不成功时主串和模式串返回的位置及 最后匹配成功函数返回的位置。

5、设二维数组A[6][10],每个数组元素占用4个存储单元，若按行优先顺序存放的数组元素，a[0][0]的存储地址为860，计算a[3][5]存储位置。

860+（3\*10+5）\*4=1000

6、画写出广义表（a,(b),((),c,d)）逐步分解为表头和表尾的过程，并画出其的存储结构。

第6章 树和二叉树

1．选择题

（1）一棵完全二叉树上有1001个结点，其中叶子结点的个数是（ D ）。

A．250 B． 500 C．254 D．501

答案：D

解释：设度为0结点（叶子结点）个数为A，度为1的结点个数为B，度为2的结点个数为C，有A=C+1，A+B+C=1001，可得2C+B=1000，由完全二叉树的性质可得B=0或1，又因为C为整数，所以B=0，C=500，A=501，即有501个叶子结点。

（2）一个具有1025个结点的二叉树的高h为（ C ）。

A．11 B．10 C．11至1025之间 D．10至1024之间

答案：C

解释：若每层仅有一个结点，则树高h为1025；且其最小树高为 ⎣log21025⎦ + 1=11，即h在11至1025之间。

（3）深度为h的满m叉树的第k层有（ A ）个结点。(1=<k=<h)

A．mk-1  B．mk-1 C．mh-1 D．mh-1

答案：A

解释：深度为h的满m叉树共有mh-1个结点，第k层有mk-1个结点。

（4）利用二叉链表存储树，则根结点的右指针是（C ）。

A．指向最左孩子 B．指向最右孩子 C．空 D．非空

答案：C

解释：利用二叉链表存储树时，右指针指向兄弟结点，因为根节点没有兄弟结点，故根节点的右指针指向空。

（5）设F是一个森林，B是由F变换得的二叉树。若F中有n个非终端结点，则B中右指针域为空的结点有（ c ）个。

A．n−1 B．n C．n + 1 D．n + 2

2．应用题

（1）设一棵二叉树的先序序列： A B D F C E G H ，中序序列： B F D A G E H C

①画出这棵二叉树，写出后序序列。（提示根据先序 和中序 逐个判断出根和左右结点）

②画出这棵二叉树的后序线索树。

③将这棵二叉树转换成对应的树（或森林）。



|  |
| --- |
| A |

|  |
| --- |
| B |

F

|  |
| --- |
| D |

|  |
| --- |
| ③ |

|  |
| --- |
| C |

|  |
| --- |
| E |

H

G

① ②

（2） 假设用于通信的电文仅由8个字母组成，字母在电文中出现的频率分别为0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.10。试为这8个字母设计赫夫曼编码。

答案：

先将概率放大100倍，以方便构造哈夫曼树。

w={7,19,2,6,32,3,21,10}，按哈夫曼规则：【[（2,3），6], (7,10)】, ­……19, 21, 32

**（100）**

**（40） （60）**

**19 21 32 （28）**

**（17） （11）**

**7 10 6 （5）**

**2 3**

0 1

0 1 0 1

1. 21 32
2. 1

0 1 0 1

1. 10 6
2. 1

2 3

方案比较：

|  |  |  |
| --- | --- | --- |
| 字母编号 | 对应编码 | 出现频率 |
| 1 | 1100 | 0.07 |
| 2 | 00 | 0.19 |
| 3 | 11110 | 0.02 |
| 4 | 1110 | 0.06 |
| 5 | 10 | 0.32 |
| 6 | 11111 | 0.03 |
| 7 | 01 | 0.21 |
| 8 | 1101 | 0.10 |

（3）算法题：假设二叉树采用二叉链表存储结构存储，设计一个算法，计算一棵给定二叉树的所有叶子结点的个数。（使用递归算法）

[题目分析]如果二叉树为空，返回0，如果二叉树不为空且左右子树为空，返回1，如果二叉树不为空，且左右子树不同时为空，返回左子树中叶子节点个数加上右子树中叶子节点个数。

[算法描述]

int LeafNodeCount(BiTree T)

{

if(T==NULL)

return 0; //如果是空树，则叶子结点个数为0

else if(T->lchild==NULL&&T->rchild==NULL)

return 1; //判断结点是否是叶子结点（左孩子右孩子都为空），若是则返回1

else

return LeafNodeCount(T->lchild)+LeafNodeCount(T->rchild);

}

第7章 图

1．选择题

（1）在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和的（ ）倍。

A．1/2 B．1 C．2 D．4

答案：B

解释：有向图所有顶点入度之和等于所有顶点出度之和。

（2）具有n个顶点的有向图最多有（ ）条边。

A．n B．n(n-1) C．n(n+1) D．n2

答案：B

解释：有向图的边有方向之分，即为从n个顶点中选取2个顶点有序排列，结果为n(n-1)。

（3）下面（　）算法适合构造一个稠密图G的最小生成树。

A． Prim算法 B．Kruskal算法 C．Floyd算法 D．Dijkstra算法

答案：A

解释：Prim算法适合构造一个稠密图G的最小生成树，Kruskal算法适合构造一个稀疏图G的最小生成树。

（4）用邻接表表示图进行广度优先遍历时，通常借助（ ）来实现算法。

A．栈 B. 队列 C. 树 D．图

答案：B

解释：广度优先遍历通常借助队列来实现算法，深度优先遍历通常借助栈来实现算法。

（5）已知图的邻接矩阵下图1所示，则从顶点*v*0出发按深度优先遍历的结果是（ C ）。

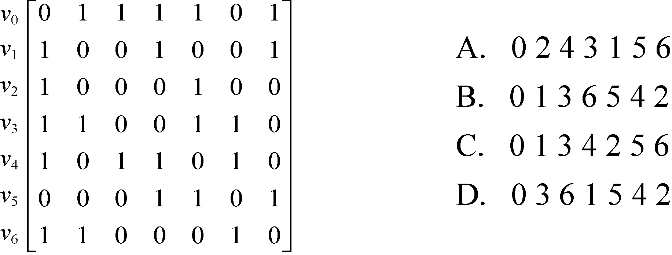


  图1 邻接矩阵

（6）已知图的邻接表如下图2所示，则从顶点*v*0出发按广度优先遍历的结果是（ D ），按深度优先遍历的结果是（ D ）。

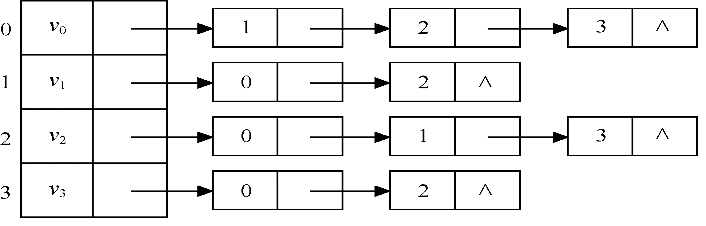


图2 邻接表

A．0 1 3 2 B．0 2 3 1 C．0 3 2 1 D．0 1 2 3

（7）下面（ ）方法可以判断出一个有向图是否有环。

A．深度优先遍历 B．拓扑排序 C．求最短路径 D．求关键路径

答案：B

**2．应用题**

（1）已知图3所示的有向图，请给出：

① 每个顶点的入度和出度；

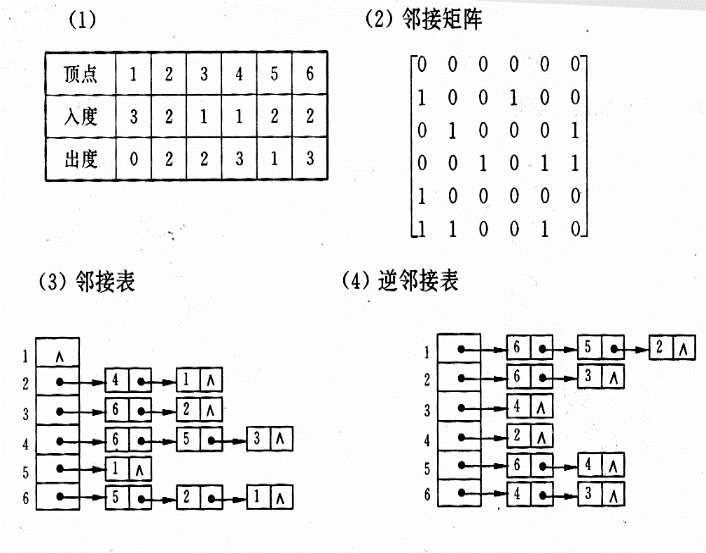
② 邻接矩阵；

③ 邻接表；

④ 逆邻接表。



  图3 有向图



（2）已知如图4所示的无向网，请用普里姆算法给出从A开始的最小生成树

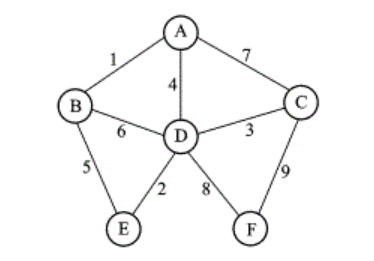


图4 无向网

（3）无网如图5所示，试用迪杰斯特拉算法求出从顶点u到其他各顶点间的最短路径，列表写出计算过程。

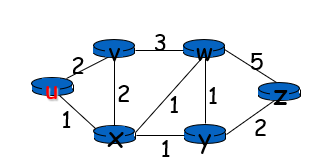


图5

（5）算法题：假设图G采用邻接表存储，分别实现图中G中每个顶点的入度的算法、 出度的算法。

第9章 查找

1．选择题

（1）对n个元素的表做顺序查找时，若查找每个元素的概率相同，则平均查找长度为（ C ）。

A．(n-1)/2 B． n/2 C．(n+1)/2 D．n

（2）适用于折半查找的表的存储方式及元素排列要求为（D ）。

A．链接方式存储，元素无序 B．链接方式存储，元素有序

C．顺序方式存储，元素无序 D．顺序方式存储，元素有序

（3）下面关于哈希查找的说法，正确的是（ C ）。

A．哈希函数构造的越复杂越好，因为这样随机性好，冲突小

B．除留余数法是所有哈希函数中最好的

C．不存在特别好与坏的哈希函数，要视情况而定

D．哈希表的平均查找长度有时也和记录总数有关

（4）下面关于哈希查找的说法，不正确的是（ A ）。

A．采用链地址法处理冲突时，查找一个元素的时间是相同的

B．采用链地址法处理冲突时，若插入规定总是在链首，则插入任一个元素的时间是相同的

C．用链地址法处理冲突，不会引起二次聚集现象

D．用链地址法处理冲突，适合表长不确定的情况

（5）设哈希表长为14，哈希函数是H(key)=key%11，表中已有数据的关键字为15，38，61，84共四个，现要将关键字为49的元素加到表中，用二次探测法解决冲突，则放入的位置是（ D ）。

A．8 B．3 C．5 D．9

**2．应用题**

**（1）假定对有序表：（3，4，5，7，24，30，42，54，63，72，87，95）进行折半查找，试回答下列问题：**

**① 画出描述折半查找过程的判定树；**

**② 若查找元素54，需依次与哪些元素比较？**

**③ 若查找元素90，需依次与哪些元素比较？**

**④ 假定每个元素的查找概率相等，求查找成功时的平均查找长度。**

答案：

**①**先画出判定树如下（注：mid=⎣(12)/2⎦=6）：

**30**

**5 63**

**3 7 42 87**

**4 24 54 72 95**

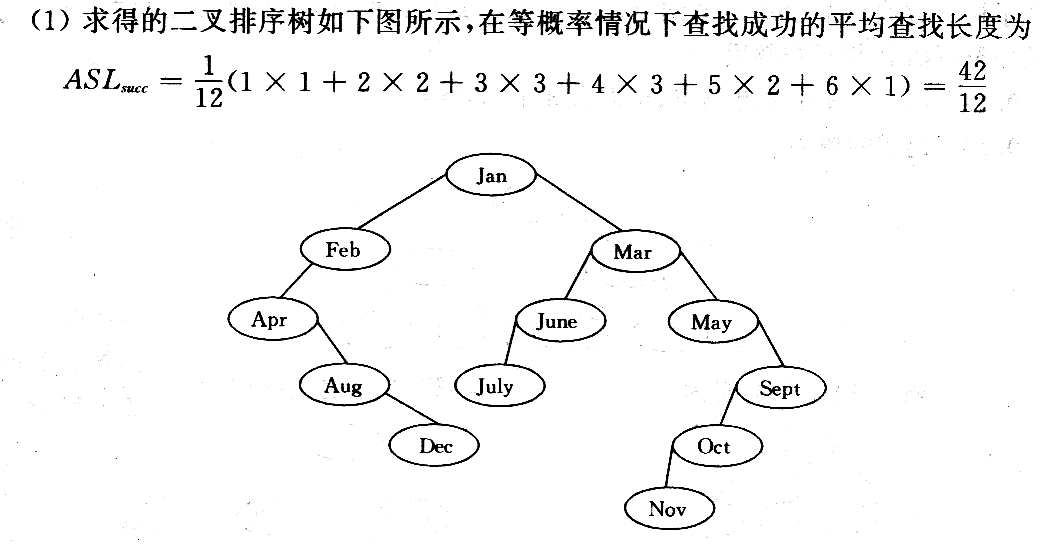
**②**查找元素54，需依次与30, 63, 42, 54 元素比较；

**③**查找元素90，需依次与30, 63,87, 95元素比较；

**④**求ASL之前，需要统计每个元素的查找次数。判定树的前3层共查找1＋2×2＋4×3=17次；

但最后一层未满，不能用8×4，只能用5×4=20次，

所以ASL＝1/12（17＋20）＝37/12≈3.08

**（2）已知如下所示长度为12的表：（Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec）试按表中元素的顺序依次插入一棵初始为空的二叉排序树，画出插入完成之后的二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。**

（**3）设哈希函数H（K）=3 K mod 11，哈希地址空间为0～10，对关键字序列（32，13，49，24，38，21，4，12），按下述两种解决冲突的方法构造哈希表，并分别求出等概率下查找成功时平均查找长度ASL**

**① 线性探测法；**

**② 链地址法。**

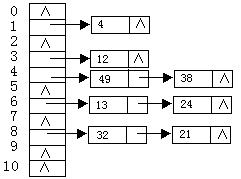
答案：

**①**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 关键字 |  | 4 |  | 12 | 49 | 38 | 13 | 24 | 32 | 21 |  |
| 比较次数 |  | 1 |  | 1 | 1 | 2 | 1 | 2 | 1 | 2 |  |

ASLsucc =（1+1+1+2+1+2+1+2）/8=11/8

**②**



ASLsucc =（1\*5+2\*3）/8=11/8

第10章 排序

1．选择题

（1）从未排序序列中依次取出元素与已排序序列中的元素进行比较，将其放入已排序序列的正确位置上的方法，这种排序方法称为（ C ）。

A．归并排序 B．冒泡排序 C．插入排序 D．选择排序

（2）从未排序序列中挑选元素，并将其依次放入已排序序列（初始时为空）的一端的方法，称为（D ）。

A．归并排序 B．冒泡排序 C．插入排序 D．选择排序

（3）快速排序在下列（ C ）情况下最易发挥其长处。

A．被排序的数据中含有多个相同排序码

B．被排序的数据已基本有序

C．被排序的数据完全无序

D．被排序的数据中的最大值和最小值相差悬殊

（4）对n个关键字作快速排序，在最坏情况下，算法的时间复杂度是（ B ）。

A．O(n) B．O(n2) C．O(nlog2n) D．O(n3)

（5）若一组记录的排序码为（46, 79，56，38，40，84），则利用快速排序的方法，以第一个记录为基准得到的一次划分结果为（ C ）。

A．38，40，46，56，79，84 B．40，38，46，79，56，84

C．40，38，46，56，79，84 D．40，38，46，84，56，79

（6）下列关键字序列中，（ D ）是堆。

A．16，72，31，23，94，53 B．94，23，31，72，16，53

C．16，53，23，94，31，72 D．16，23，53，31，94，72

（7）堆是一种（ B）排序。

A．插入 B．选择 C．交换 D．归并

（8）堆的形状是一棵（ C ）。

A．二叉排序树 B．满二叉树 C．完全二叉树 D．平衡二叉树

（9）若一组记录的排序码为（46，79，56，38，40，84），则利用堆排序的方法建立的初始堆为（ B ）。

A．79，46，56，38，40，84 B．84，79，56，38，40，46

C．84，79，56，46，40，38 D．84，56，79，40，46，38

（10）下述几种排序方法中，（ C ）是稳定的排序方法。

A．希尔排序 B．快速排序 C．归并排序 D．堆排序

2．应用题

设待排序的关键字序列为{12，2，16，30，28，10，16\*，20，6，18}，试分别写出使用以下排序方法，每趟排序结束后关键字序列的状态，并写出最后排序结果。

① 直接插入排序

② 折半插入排序

③ 希尔排序（增量选取5，3，1）

④ 冒泡排序

⑤ 快速排序

⑥ 简单选择排序

⑦ 堆排序

⑧ 二路归并排序

答案：

①直接插入排序

[2 12] 16 30 28 10 16\* 20 6 18

[2 12 16] 30 28 10 16\* 20 6 18

[2 12 16 30] 28 10 16\* 20 6 18

[2 12 16 28 30] 10 16\* 20 6 18

[2 10 12 16 28 30] 16\* 20 6 18

[2 10 12 16 16\* 28 30] 20 6 18

[2 10 12 16 16\* 20 28 30] 6 18

[2 6 10 12 16 16\* 20 28 30] 18

[2 6 10 12 16 16\* 18 20 28 30]

② 折半插入排序 排序过程同①

③ 希尔排序（增量选取5，3，1）

10 2 16 6 18 12 16\* 20 30 28 （增量选取5）

6 2 12 10 18 16 16\* 20 30 28 （增量选取3）

2 6 10 12 16 16\* 18 20 28 30 （增量选取1）

④ 冒泡排序

2 12 16 28 10 16\* 20 6 18 [30]

2 12 16 10 16\* 20 6 18 [28 30]

2 12 10 16 16\* 6 18 [20 28 30]

2 10 12 16 6 16\* [18 20 28 30]

2 10 12 6 16 [16\* 18 20 28 30]

2 10 6 12 [16 16\* 18 20 28 30]

2 6 10 [12 16 16\* 18 20 28 30]

2 6 10 12 16 16\* 18 20 28 30]

⑤ 快速排序

12 [6 2 10] **12** [28 30 16\* 20 16 18]

6 [2] **6**  [10] 12 [28 30 16\* 20 16 18 ]

28 2 6 10 12 [18 16 16\* 20 ] **28** [30 ]

18 2 6 10 12 [16\* 16]  **18**  [20] 28 30

16\* 2 6 10 12 16\* [16] 18 20 28 30

左子序列递归深度为1，右子序列递归深度为3

⑥ 简单选择排序

2 [12 16 30 28 10 16\* 20 6 18]

2 6 [16 30 28 10 16\* 20 12 18]

2 6 10 [30 28 16 16\* 20 12 18]

2 6 10 12 [28 16 16\* 20 30 18]

2 6 10 12 16 [28 16\* 20 30 18]

2 6 10 12 16 16\* [28 20 30 18]

2 6 10 12 16 16\* 18 [20 30 28]

2 6 10 12 16 16\* 18 20 [28 30]

2 6 10 12 16 16\* 18 20 28 [30]

⑧ 二路归并排序

2 12 16 30 10 28 16 \* 20 6 18

2 12 16 30 10 16\* 20 28 6 18

2 10 12 16 16\* 20 28 30 6 18

2 6 10 12 16 16\* 18 20 28 30