

华中师范大学计算机科学系 2002--2003 年秋季

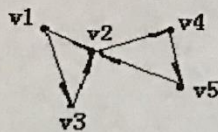
数据结构试卷

题号	一	二	三	四	五	总分	
题分	10	20	16	20	34	核分人	
得分						复查人	

一、单项选择(每题 1 分, 共 10 分)

1. 若广义表 K 满足 $\text{head}(K)=\text{tail}(K)$, 则 K 为 ()
 A. $()$ B. $(())$ C. $((), ())$ D. $((), (), ())$
2. 若要求尽可能快地对实数数组进行稳定的排序, 则应选 ()
 A. 快速排序 B. 堆排序 C. 归并排序 D. 基数排序
3. 请指出在顺序表 $\{2, 5, 7, 10, 14, 15, 18, 23, 35, 41, 52\}$ 中, 用二分法查找关键码 12 需做多少次关键码比较。 ()
 A. 2 B. 3 C. 4 D. 5
4. 对包含 N 个元素的散列表进行查找, 平均查找长度 ()
 A. 为 $O(\log_2 N)$ B. 为 $O(N)$
 C. 不直接依赖于 N D. 上述三者都不是
5. 一个栈的输入序列为 1, 2, 3, 4, 下面哪一个序列不可能是这个栈的输出序列? ()
 A. 1, 3, 2, 4 B. 2, 3, 4, 1
 C. 4, 3, 1, 2 D. 3, 4, 2, 1
6. 下面关于图的存储的叙述中, 哪一个正确的。 ()
 A. 用相邻矩阵法存储图, 占用的存储空间数只与图中结点数有关, 而与边数无关
 B. 用相邻矩阵法存储图, 占用的存储空间数只与图中边数有关, 而与结点数无关
 C. 用邻接表法存储图, 占用的存储空间数只与图中结点数有关, 而与边数无关
 D. 用邻接表法存储图, 占用的存储空间数只与图中边数有关, 而与结点数无关
7. 首先访问结点的左子树, 然后访问该结点, 最后访问结点的右子树, 这种遍历称为 ()
 A. 前序遍历 B. 后序遍历
 C. 中序遍历 D. 层次遍历
8. 对一棵查找树根结点而言, 左子树中所有结点与右子树中所有结点的关键字大小关系是 ()
 A. 小于 B. 大于 C. 等于 D. 不小于
9. 下面关于 B-树和 B+树的叙述中, 不正确的是 ()
 A. B-树和 B+树都是平衡的多分树
 B. B-树和 B+树都是可用于文件的索引结构
 C. B-树和 B+树都能有效地支持顺序检索
 D. B-树和 B+树都能有效地支持随机检索

10. 给定下列有向图和初始结点 V_1 , 按深度优先遍历的结点序列为()



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

二、填空题 (每小题 2 分, 共 20 分)

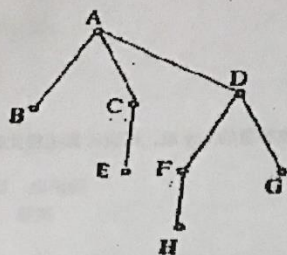
1. 从逻辑结构看, 线性表是典型的_____, 树是典型的_____。
2. 设有二维数组 $A[0..9, 0..19]$, 其每个元素占两个字节, 第一个元素的存储地址为 100, 若按行优先顺序存储, 则元素 $A[6, 6]$ 的存储地址为_____, 按列优先顺序存储, 元素 $A[6, 6]$ 的存储地址为_____。
3. 若按层次顺序将一棵有 n 个结点的完全二叉树的所有结点从 1 到 n 编号, 那么当 i 为_____且小于 n 时, 结点 i 的右兄弟是结点_____, 否则结点 i 没有右兄弟。
4. 求具有最小带权外部路径长度的扩充二叉树的算法称为_____算法。堆排序中建堆的方法称作_____。
5. 一个串, 除自身之外的所有子串都是该串的_____。
6. 在图结构中, 如果一个从 V_p 到 V_q 的路径上除 V_p 和 V_q 可以相同外, 其它结点都不相同, 则称此路径为一_____回路。
7. 树形选择排序总的时间开销为_____。
8. 6 阶 B-树中, 每个结点至多包含_____个关键码, 除根和叶结点外, 每个结点至少包含_____个关键码。
9. 散列文件是根据文件中关键字的特点设计一种_____函数和_____方法将记录散列到存储器上的文件。
10. 磁带和磁盘中, _____适合随机存储, _____适合顺序存储。

三、简答题(每小题 4 分, 共 16 分)

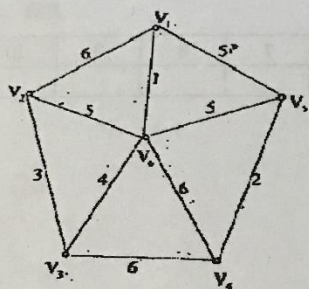
1. 设有 K 个关键字互为同义词, 若用线性探测法把这 K 个关键字存入散列表中, 至少要进行多少次探测?
2. 什么是二叉排序树? 什么是二叉平衡树?
3. 一棵树有度为 1 的结点 n_1 个, 度为 2 的结点 n_2 个, ..., 度为 m 的结点 n_m 个, 问它有多少个叶结点?
4. 什么是散列表的装填因子? 为什么说当装填因子非常接近 1 时, 线性探查类似于顺序查找? 为什么说当装填因子比较小(比如 $\alpha=0.7$ 左右)时, 散列查找的平均查找时间为 $O(1)$?

四、应用题: (每题 5 分, 共 20 分)

1. 把下面的树变成二叉树。



2. 在一棵空的二叉查找树中依次插入关键字序列为 20、30、8、12、34、5、60、5、1、29，请画出所得到的二叉查找树。
3. 画出下列网络的最小生成树。



4. 假设用于通信的电文仅由 A-H 八个字母组成，字母在电文中出现的频率分别为 7，19，2，6，32，3，21，10。试为这八个字母设计哈夫曼编码。

五、算法题(共 34 分)

1. 试写一算法写出用二叉链表表示给定二叉树的叶结点总数。(12 分)
2. 下面给出了冒泡排序算法，请填写算法中的空框，使算法正确。(10 分)

```

typedef struct { int key;
                 datatype info;           //设 datatype 已经定义
               } node;

void BubbleSort ( node R[]) //R 中元素个数为 n 个
{
    node X;
    i=L.length;
    While(i (1) 1)
    {
        LastExchangeIndex= (2) ;
        For(j=1; j<i; j++)
        {
            if(L.r[j+1].key (3) r[j].key)
            {
                X=L.r[j]; L.r[j]=L.r[j+1]; L.r[j+1]=X;
                lastEXchangeIndex= (4) ;
            }
        }
    }
}
  
```

```

    }
    (5) = lastEXchangeIndex;
}
} // 算法结束

```

3、设一单链表的头指针为 head,链表的结点中包含着整数类型的 key 域,试设计算法将此链表的结点按照 key 递增次序进行就地排序。
(12 分)

答卷纸:

题号	一	二	三	四	五	总分	
题分	10	20	16	20	34	核分人	
得分						复查人	

一、选择题

1	2	3	4	5	6	7	8	9	10

二、填空

- 1、 _____
- 2、 _____
- 3、 _____
- 4、 _____
- 5、 _____
- 6、 _____
- 7、 _____
- 8、 _____
- 9、 _____
- 10、 _____

三、简答题

1、.答:

2、.答:

3、.答:

4、答:

四、应用题

1、答案

2、答案

3、答案

4、答案

五、算法题

1、答案:

王道论坛cskaoyan.com

2、答案:

- 1) _____
- 2) _____
- 3) _____
- 4) _____
- 5) _____

3、答案:

王道论坛cskaoyan.com

答卷纸:

题号	一	二	三	四	五	总分	
题分	10	20	16	20	34	核分人	
得分						复查人	

一、选择题

1	2	3	4	5	6	7	8	9	10
B	C	C	C	C	A	C	A	D	A

二、填空

- 1、线性结构 非线性结构
- 2、352 232
- 3、偶数 $i+1$
- 4、迪杰斯特拉 筛选法
- 5、真子串
- 6、简单路径 简单回路
- 7、 $O(n \lg n)$
- 8、5 2
- 9、哈希函数 处理冲突
- 10、磁盘 磁带

三、简答题

1、(答案及点评) 1.答: 至少要进行 $K(K+1)/2$ 次探测。

假设全部是同义词, 存第一个关键字时进行 1 次探测, 存第二个时至少要进行 2 次探测... 存第 K 个时要进行 K 次探测, 因此总共至少为上述次数。

2、(答案及点评) 答: 二叉排序树又称二叉查找树。它或者是空树, 或者是符合下列条件的二叉树:

- (1) 如果它的左子树非空, 则它的左子树上全部结点的值均小于根结点的值。
- (2) 如果它的右子树非空, 则它的右子树上全部结点的值均大于根结点的值。
- (3) 它的左、右子树分别又是一棵二叉排序树。

平衡二叉树是指其任一结点上的左子树和右子树的高度大致相等的二叉树。

3、(答案及点评) 假设根结点度为零时, 叶结点数为 1; 若树中每出现一个度为 1 的结点, 则叶子数增加 0 个, 每出现 1 个度为 2 的结点, 叶子数增加 1 个, 每出现 1 个度为 3 的结点, 则叶子数增加 2 个... 依此类推, 每个度为 m 的结点, 可增加 $m-1$ 个叶子数。因此可回答:

答: 此树的叶结点总数应为:

$$\begin{aligned} n_0 &= 1 + 0 \cdot n_1 + 1 \cdot n_2 + 2 \cdot n_3 + \dots + (m-1) \cdot n_m \\ &= 1 + n_2 + 2n_3 + \dots + (m-1)n_m \quad (\text{个}) \\ n_0 &= 1 + n_1(1-1) + (n_2-1) \end{aligned}$$

4、(答案及点评) 4.答: 散列表的装填因子是指散列表中装填的结点数 n 与散列表的长度 m 的比值 n/m 。

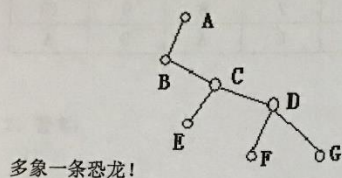
当 α 非常接近 1 时, 整个散列表几乎被装满。由于线性探查法在关键字同义时解决冲突

的办法是线性地向后查找，当整个表几乎装满时，它就类似于顺序查找了。

当 α 比较小时，关键字碰撞的几率比较小，一般情况下只要按照散列函数计算出的结果能够 1 次性就找到相应结点，因此它的平均查找时间为 $O(1)$ 。

四、应用题

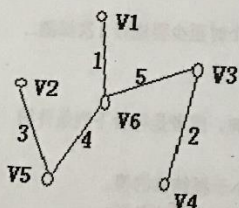
1、(答案及点评) 答案见下图。



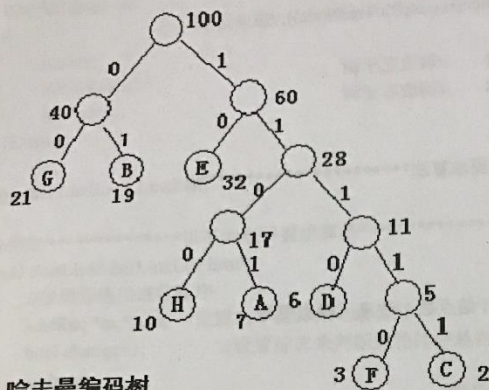
2、(答案及点评)



3、(答案及点评)



4、(答案及点评) 如图所示，由哈夫曼编码树可得各字母的编码如下：



哈夫曼编码树

A: 1101
B: 01
C: 11111
D: 1110
E: 10
F: 11110
G: 00
H: 1100

五、算法题

1、求叶结点数问题 (答案及点评)

```
#include <stdio.h>
```

```
#include <malloc.h>
```

```
typedef char Datatype; // 以下为二叉链表的结构定义
```

```
typedef struct node
```

```
{
```

```
    Datatype data;
```

```
    node *lchild;
```

```
    node *rchild;
```

```
} BinTreeNode;
```

```
typedef BinTreeNode *BinTree;
```

```
void CreatBinTree(BinTree *T)
```

```
{    // 构造二叉链表。T 是指向根的指针，故修改了*T 就修改了实参
```

```
    char ch;
```

```
    if ((ch=getchar())!='')
```

```
        *T=NULL;
```

```

else{ //读入非空格
    *T=(BinTreeNode *)malloc(sizeof(BinTreeNode)); //生成结点
    (*T)->data=ch;
    CreatBinTree(&(*T)->lchild); //构造左子树
    CreatBinTree(&(*T)->rchild); //构造右子树
}
}
//*****以下为题目要求算法*****//

int GetLeaves( BinTree root)
{
    //求叶结点总数
    static int leaf=0; //此1用于记叶结点数, 注意用静态变量
    if(root)
    {
        //递归计算叶结点数
        if(!root->lchild||root->rchild)
            leaf++; //如果该结点无左右孩子, 则叶子数加1
        GetLeaves(root->lchild); //算左子树的叶结点数
        GetLeaves(root->rchild); //算右子树的叶结点数
    }
    return leaf; //返回结果
}
//*****算法结束*****//

void main()
{
    //以下为验证程序
    BinTree root;
    CreatBinTree(&root);
    int leaves=GetLeaves(root);
    printf("Total leaves=%d",leaves);
}

```

2、(答案及点评)

- 1) >
- 2) 1
- 3) <
- 4) j
- 5) i

3、(答案及点评)

单链表排序

```
#include <stdio.h>
```

```
#include <malloc.h>
```

```
typedef char RecInfo; //单链表结构定义
```

```

typedef struct rec
{
    int key;
    RecInfo info;
    rec * next;
}ListRec;

typedef ListRec * LinkList;

//////////*****以下为题目要求算法*****//////////
void SortLinkList(LinkList head)
{
    //采用冒泡法进行排序
    ListRec *m,*p,*s; //设置三个指针来记录位置
    bool change=1;      //设置标志来判断此轮排序是否进行交换
    m=head;
    p=head->next;
    s=p->next;
    while (m&&change)
    {
        //外循环用于判断和进行第 X 轮排序
        change=0;      //标志进行复位
        while(p&& s)
        {
            //以下为比较两个结点 key 值大小，若前者更大，则进行交换
            if(p->key<=s->key)
            {
                m=p;
                p=s;
                s=s->next;
            }
            else
            {
                p->next=s->next;
                s->next=p;
                m->next=s;
                change=1;
                m=s;s=p->next;
            }
        }
        m=head;      //此处三行将指针复位
        p=head->next;
        s=p->next;
    }
}
//*****算法结束

```

//其他语句都是用来验证算法用的


```

LinkedList CreatList( void)
{
    char ch;
    LinkedList head=(ListRec *)malloc(sizeof(ListRec));
    ListRec *s,*r;
    r=head;
    while((ch=getchar())!='\n')
    {
        s=(ListRec *)malloc(sizeof(ListRec));
        s->info=ch;
        s->key=ch;
        r->next=s;
        r=s;
    }
    r->next=NULL;
    return head;
}

void ShowLinkedList(LinkedList head)
{
    ListRec *h=head->next;
    while(h)
    {
        printf("%c ",h->info);
        h=h->next;
    }
}

void main()
{
    //验证程序，你可以输入各种字符序列来验证
    LinkedList head=CreatList();
    SortLinkedList(head);
    ShowLinkedList(head);
}

```