**数据结构试卷**

1. **判断题（5’）**

1.单链表中的头结点就是单链表的第一个结点。（错）

2.栈的修改是按后进先出的原则进行的。 √

3.任何一个非空列表，其表头可以是原子，也可以是列表，表尾也是如此（错）

4.查找效率最高的二叉排序树是平衡二叉树。 （对）

5.在一颗平衡二叉树，每个结点的平衡因子的取值范围是一1和1之间。 （对）

1. **填空题（10’）**

1.数据结构是由4类基本结构组成，分别是集合，线性结构，树状结构，图状结构或网状结构。

2.算法的设计要求为正确性，可读性，健壮性，高效性。

3. 在单链表中设置头结点的作用是（简化插入删除算法）

4.在一个带头结点的单循环链表中，p指向尾结点的直接前驱，则指向头结点的指针head可用p表示为（p—> next—>next=head）

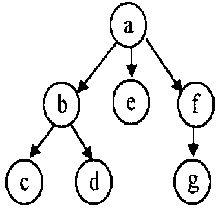
5. 判定一个循环队列QU（最多元素为m，m==MAXSIZE-1）为满队列的条件是QU->front==(QU->rear+1)%MAXSIZE;

6. 除了结构的初始化和销毁之外，数组只有存取元素和修改元素的操作。

7. 假设在m\*n的矩阵中，有t个元素不为0，则稀疏因子k为k=t/(m\*n)。

8. 对于一个具有n个顶点的无向完全图中，包含有 n(n-1)/2 条边，在一个具有n个顶点的有向完全图中，包含有 n(n-1) 条边。

9. 已知一个图的广度优先生成树如右图所示，则与此相应的广度优先遍历序列为 abefcdg



10.高度为8的平衡二叉树的结点数至少有54个。

1. **选择题（10’）**

1.程序：

for (i=1;i<=n;++i) {++x;s+=x;}

for (j=1;j<=n;++j)

{

for(k=1;k<=n;++k) {++x;s+=x;}

}

的时间复杂度为T(m)=( )

1. O(m) B. O(m) C. O(㎡) D. O(㏒m)

2.一个向量第一个元素的顺序存储地址是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、104      D、105

3. 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

4. 栈的基本操作出了在B进行插入或删除外，还有栈的初始化，判断是否为空级取栈顶元素等

A．栈底 B.栈顶 C.栈头 D.栈尾

5. 已知二维数组A[m,n],第一个元素的存储地址为loc（0,0），每个元素占k个存储单元，则以下哪个表达式可以正确表示loc（i,j）？（D）

A、loc（i,j）=loc（0,0）+（m\*j+i）\*k

B、loc（i,j）=loc（0,0）+（m\*i+j）\*k

C、loc（i,j）=loc（0,0）+（n\*j+i）\*k

D、loc（i,j）=loc（0,0）+（n\*i+j）\*k

6.算术表达式a+b\*(c+d/e)转为后缀表达式后为（ B ）。（4分）

A.ab+cde/\* B.abcde/+\*+

C.abcde/\*++ D.abcde\*/++

4、在下列情况中，可称为二叉树的是（ B ）。（4分）

A、每个结点至多有两棵子树的树            B、哈夫曼树

C、每个结点至多有两棵子树的有序树        D、每个结点只有一棵右子树

E、以上答案都不对

7.一棵有124个叶结点的完全二叉树，最多有（A ）个结点。（4分）。

A、247    B、248    C、249    D、250

8.设有6个结点的无向图，该图至少应有（A）条边才能确保是一个连通图。

A. 5 B .6 C .7 D .8

9.假设有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（D）。

A、K一1次 B 、K次 C、K+1次 D、K（K+1）/2次

10.关于B一和B+树的叙述中，不正确的是（C）。

A、B一和B+树都是平衡的多叉树

B、B一和B+树都可用于文件的索引结构

C、B一和B+树都能有效的支持顺序检索

D、B一和B+树都能有效地支持随机检索

1. **读下列程序，写出实现了什么算法功能（2\*5’）**

**1.下述算法的功能是什么?**  
　　LinkList Demo(LinkList L){ // L 是无头结点单链表  
　　　ListNode \*Q,\*P;  
　　　if(L&&L->next){  
　　　　Q=L;L=L->next;P=L;  
　　　　while (P->next) P=P->next;  
　　　　　P->next=Q; Q->next=NULL;  
　　　　}  
　　　　return L;  
　　}// Demo  
  
**答**：  
　　该算法的功能是：将开始结点摘下链接到终端结点之后成为新的终端结点，而原来的第二个结点成为新的开始结点，返回新链表的头指针。

2. void BubbleSort

//endfor

exchange=FLASE; file://重 新设置标记

for file://对 当前无序区R进行自上而下扫描

if return; file://本 趟未发生交换，提前结束

}//endfor

}//endfor

}//BubbleSort

答：一个双向冒泡排序的算法，在排序过程中交替改变扫描方

1. **程序填空题（3\*5’）**

1.向单链表的末尾添加一个元素的算法

void insertrear(LNode\* HL,const ElemType &item)

{

LNode\* newptr;

newptr=new LNode;

if(newptr==NULL)

{

cerr<<"Memory allocation failare!"<<endl;

exit(1);

}

newptr->data=item;

newptr->next=NULL;

if(HL==NULL)

HL=newptr;

else{

LNode\* p=HL;

while (p->next!=NULL)

{

p=p->next;

}

p->next=newptr;

}

}

2.二分查找的递归算法

int Binsch(ElemType a[], int low, int high, KeyType k)

{

if(low<=high)

{

int mid=(low+high)/2;

if(k==a[mid].key) return mid;

else if(k<a[mid].key)

return Binsch(a,low,mid-1,k);

else

return Binsch(a,mid+1,high,k);

}

else

return -1;

}

3.二叉树的查找——递归算法

bool Find(BTreeNode\* BST, Elemtype &item)

{

if(BST==NULL)

return false; //查找失败

else{

if(item==BST->data){

item = BST->data;

return true;} //查找成功

else if (item<BST->data)

{

return Find(**BST->left**, item);

}

else

return Find(BST->right, item);

}

}

1. **算法实现题（2\*10’）**

1.编写递归算法，将二叉树中所有结点的左、右子树相互交换。（10分）

void Bitree\_Revolute(Bitree T)//交换所有结点的左右子树

{

  T->lchild<->T->rchild; //交换左右子树

  if(T->lchild) Bitree\_Revolute(T->lchild);

  if(T->rchild) Bitree\_Revolute(T->rchild); //左右子树再分别交换各自的左右子树

}//Bitree\_Revolute

2.试编写算法，对一棵以孩子-兄弟链表表示的树统计叶子的个数。（10分）

int LeafCount\_CSTree(CSTree T)//求孩子兄弟链表表示的树T的叶子数目

{

  if(!T->firstchild) return 1; //叶子结点

  else

  {

count=0;

    for(child=T->firstchild;child;child=child->nextsib)

      count+=LeafCount\_CSTree(child);

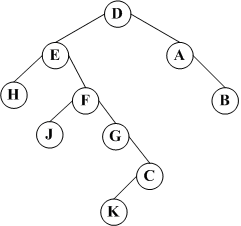
    return count; //各子树的叶子数之和

  }

}//LeafCount\_CSTree

1. **简答题（4\*5’）**

**1**.根据下图给出的树写出以下遍历序列：（1）先根序列；（2）后根序列。（5分）



（1）DEHFJGCKAB （2）HJKCGFEBAD

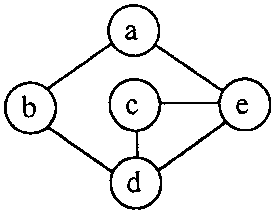
**2.**已知一个无向图的顶点集为｛a,b,c,d,e｝，其邻接矩阵如下所示：

1. 1 0 0 1 a
2. 0 0 1 0 b
3. 0 0 1 1 c

0 1 1 0 1 d

1. 0 1 1 0 e
2. 画出该图的图形；

该图的图形为： 3’

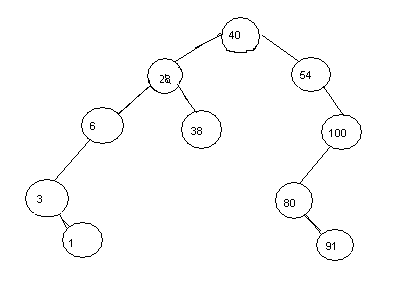
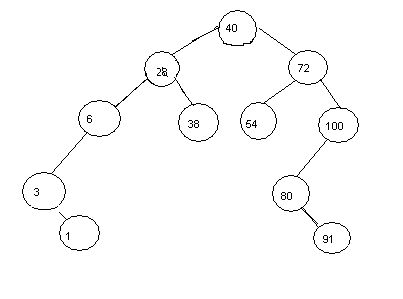
答：

（2）根据邻接矩阵从顶点a出发进行深度优先遍历和广度优先遍历，写出相应的遍历序列。

深度优先遍历序列为：abdce 1’

广度优先遍历序列为：abedc。 1’

3. 输入一个正整数序列｛40，28，6，72，100，3，54，1，80，91，38｝，建立一个二叉排序树，然后删除结点72，分别画出该二叉树及删除结点72后的二叉树。 5’



4．有一个2000项的表，要采用等分区间顺序查找的分块查找法，问：

（1）每块理想长度是多少？ 1’

（2）分成多少块最为理想？ 1’

（3）平均查找长度为多少？ 1’

（4）若每块是20，则平均查找长度为多少？ 2’

答：（1）理想长度是d=2000^(1/2) =45块

（2）设d为块长，长度为n 的表分成b=[n/d],故有b=[n/d] =[2000/45].

(3)因块查找和块查找都采用顺序查找法，所以：

ASL =（b+1）/2+(d+1)/2=46

(4)每块的长度为20，b=[n/d] = [2000/20] =100块，所以ASL=（b+1）/2+(d+1)/2=61