重庆理工大学考试试卷

一、单项选择(每题2分，共20分)

1．一个具有n个顶点的无向完全图的边数为（ ）

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

2．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.连续不连续都可以

3．一个顺序表第一个元素的存储地址是 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.100 D.120

4．串是一种特殊的线性表，其特殊性体现在（ ）

A.可以顺序存储 B.数据元素是一个字符

C.可以链接存储 D.数据元素可以是多个字符

5．顺序查找法适合于存储结构为（ ）的线性表。

A.散列存储 B.顺序存储或链接存储

C.压缩存储 D.索引存储

6．下述几种排序方法中，平均时间复杂度最小的是（ ）

A.插入排序 B.选择排序 C.冒泡排序 D.快速排序

7. 循环队列用数组 A[0，m-1]存放其元素值，已知其头尾指针分别是 front 和 rear，则当前队列中的元素个数是 。

A. (rear-front+m) % m B. read-front+1

C. read-front-1 D. read-front

8.3个结点可构成（ ）个不同形态的二叉树。

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

9．对于关键字值序列（12，13，11，18，60，15，7，18，25，100），用筛选法建堆，必须从关键字值为（ ）的结点开始。

A.100 B.12 C.60 D.15

10．栈操作的原则是（ ）

A.先进先出 B.后进先出 C.栈顶插入 D.栈顶删除

二、填空题（每小题1分，共10分），将正确的答案写在每小题的空格内。

1. 在双循环链表中，在指针p所指结点前插入指针s所指的结点，需执行下列语句：s->next=p;s->prior=p->prior;p->prior=s; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_=s; 2．散列法存储中处理碰撞（冲突）的两类基本方法是 、

3. 设二维数组A[-2..10,-1..20]按行优先顺序存储，每个元素占4个存储单元，A[-2,-1]的存储地址是1000,则A[5,6]的存储地址是 。

4. 设s[1..maxsize]为一个顺序存储的栈，变量top指示栈顶位置，栈为空的条件是\_\_\_\_\_\_\_\_\_\_\_，栈为满的条件是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

5． 设有向图的邻接矩阵为A，如果图中不存在弧VI，VJ，则A[I][J]的值为： 。

6. 设有一个链队列，结点结构为：队尾指针为Ls（≠null），则执行入队操作时， S->next=Ls->next；\_ \_\_\_\_\_\_\_\_\_\_；\_ \_\_\_\_\_\_\_\_\_\_。 7. 单链表中指针P所指结点不为尾结点的条件是\_ \_\_\_\_\_\_\_\_\_\_。

8. G为无向图，如果从q的某个顶点出发，进行一次广度优先搜索，即可访问图的每个顶点，则该图一定是 图。

9. 串S=″I am a worker″的长度是\_ \_\_\_\_。 10.按先根次序法遍历树林正好等同于按 遍历对应的二叉树。

1

5 5 3

2 4

三、算法应用题(每小题5分,共20分)

1、 对上图所示的网，执行prim算法可得到最小生成树，试在下表的空白处填上适当的内容，以说明该算法的执行过程。

顶点 1 3 4 U V

（U，V）（2，3）（2，4）{2} 3，代价 ∞ 4 2

（U，V）（2，3） {2，4} {1，3} 代价 4

（U，V） {2，4，3} {1} 代价 1

（U，V） 代价

2、有一组关键码序列{8，9，5，3，7，2，1}，分别采用冒泡排序、快速排序、直接选择排序、直接插入排序、二路归并排序方法由小到大进行排序，在下面的选项中选择各种排序第一趟排序的结果。

冒泡排序： ；快速排序： ；直接选择排序： 直接插入排序： ；二路归并：

A．{1，2，5，3，7，8，9}

B．{1，9，5，3，7，2，8}

C．{9，8，5，3，7，2，1}

D．{9，5，3，7，2，1，8}

E．{8，5，3，7，2，1，9}

F．{8，9，3，5，2，7，1}

3、设有序序列30,18,3,61,14,49，请按该序列构成一棵二叉排序树，并求查找3的比较次数。

，它们的权值分别为{5，10，12，15，30}，4、现有5个结点（A，B，C，D，E）

在下面的选项中选择一个编号，说明这5个结点的哈夫曼编码。（ ）

（1）A：1，B：001，C：010，D：011，E：000

（2）A：000，B：001，C：010，D：011，E：1

（3）A：001，B：011，C：010，D：000，E：1

（4）A：000，B：1，C：010，D：011，E： 001

四、算法填空和分析（每小题5分，共30分）

1、设单链表存放字符串,下列算法使用栈判断该字符串是否是中心对称,如abccba即为中心对称字符串. 根据题目填空完善程序.

typedef struct node

{

char data；

struct node \*next；

}cnode；

int judge(cnode \*h)

{

char st[MaxLen]；

int top=0； cnode \*p= ；

while (p!=NULL)

{

st[top]=p->data；

top ；

p=p->next；

}

p= ；

while (p!=NULL)

{ top ；

if (p->data st[top])

p=p->next；

else

break；

}

if (p= =NULL)

return 1；

else

return 0；

}

2、void Merge (Elem SR[], Elem TR[], int i, int m, int n) {// 将有序的SR[i..m]和SR[m+1..n]归并为有序的TR[i..n]

for (j=m+1, k=i; i<=m && j<=n; ++k)

{ // 将SR中记录由小到大地并入TR if (SR[i].key<=SR[j].key) TR[k] = ；

else TR[k] = SR[j++]; }

if ( ) TR[k..n] = SR[i..m];// 将剩余的SR[i..m]复制到TR if ( ) TR[k..n] = SR[j..n];// 将剩余的SR[j..n]复制到TR } // Merge

3、

int Partition (Elem R[], int low, int high)

{// 交换记录子序列R[low..high]中的记录，使枢轴记录到位，并返回//其所在位置，此时，在它之前（后）的记录均不大（小）于它

R[0] = R[low];// 用子表的第一个记录作枢轴记录

pivotkey = R[low].key; // 枢轴记录关键字

while (low<high)< p=""></high)<>

{// 从表的两端交替地向中间扫描

while(low<high&& r[high].key="">=pivotkey) --high;</high&&>

R[low] = R[high];// 将比枢轴记录小的记录移到低端 while (low<="pivotkey)" ;<="" p="">

R[high] = R[low];// 将比枢轴记录大的记录移到高端

} R[low] = ; // 枢轴记录到位

return low; // 返回枢轴位置

} // Partition

4、折半查找算法：

int binsrch(JD r[],int n,int k)

{ int low,high,mid,found;

low=1; high=n; found=0;

while((low<=high)&&(found==0))

{ mid=(low+high)/2; if(k>r[mid].key) ;

else if(k==r[mid].key) found=1;

else high=mid-1;

}

if(found==1)

return( );

else

return(0);

}

5、设有一个表头为head的单链表。通过遍历一趟链表，将链表中所有结点按逆序链接。

typedef struct node

{int data;

struct node \*next;

}pointer;

Void invert(pointer head)

{p= ;

while (head!=NULL)

{u=head;

head=head->next;

u->next= ;

p=u;

}

head=p;

}

6、已知带权图的邻接矩阵表示和邻接表表示的形式说明分别如下： #define MaxNum 50//图的最大顶点数

#define INFINITY INT\_MAX //INT\_MAX为最大整数，表示∞

typedef struct{

char vexs[MaxNum];//字符类型的顶点表

int edges[MaxNum][MaxMum];//权值为整型的邻接矩阵

int n,e;//图中当前的顶点数和边数

}MGraph;//邻接矩阵结构描述

typedef struct node {

int adjvex;//邻接点域

int weight;//边的权值

struct node \*next;//链指针域

} EdgeNode;//边表结点结构描述

typedef struct {

char vertex;//顶点域

EdgeNode \* firstedge;//边表头指针

} VertexNode ;//顶点表结点结构描述

typedef struct {

VertexNode adjlist[MaxNum];//邻接表

int n,e;//图中当前的顶点数和边数

} ALGraph;//邻接表结构描述

下列算法是根据一个带权图的邻接矩阵存储结构G1建立该图的邻接表存储结构G2，请填入合适的内容，使其成为一个完整的算法。

void convertM(MGraph \*G1,ALGraph \*G2)

{

int i,j;

EdgeNode \* p;

G2->n=G1->n;

G2->e=G1->e;

for(i=0;in;i++)

{

G2->adjlist[i].vertex=G1->vexs[i];

G2->adjlist[i].firstedge= ;

}

for (i=0;in;i++)

for (j=0;jn;j++)

if(G1->edges[i][j]<infinity)< p=""></infinity)<>

{

p=(EdgeNode \*) malloc(sizeof(EdgeNode));

p->weight= ;

p->adjvex=j;

p->next=G2->adjlist[i].firstedge; ;

}

}

： 五、编写算法（每小题10分，共20分）

1、设某单链表L的结点结构如下，单链表L用于存放某人的电话薄，现在由于电话号码从7位升为8位（加60000000），请用C语言编写算法，实现电话薄中所有电话号码从7位升为8位。

Typedef struct node

{

char name[9];//姓名

long data;//电话号码

struct node \*next;

}pointer;

int change(pointer L)

{

}

2、设二叉树采用二叉链表表示，编写算法，将二叉树中所有结点的左右子树相互交换。

Typedef struct node

{

int data;

struct node \* lchild, rchild;

} \*Bitree;

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

{

}

本文件经重理工资料库整理

免费分享给广大同学

禁止任何人及机构利用本文档进行牟利。

——左洪瑜、王鑫达

2016.10.25