**考试科目**： 数据结构（A）评分标准及参考答案

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **题号** | **一** | **二** | **三** | **四** | **五** | **平时成绩** | **总分** |
| 得分 |  |  |  |  |  |  |  |
| 评卷  教师 |  | | | | | | |

|  |  |
| --- | --- |
| **得分** |  |

**一、单项选择题（在每个小题四个备选答案中选出一个正确答案，填在题末的括号中）（本大题共10小题，每小题1.5分，总计10分）(选对1个题给1.5分，选错1个题不给分)**

1、从逻辑上可以把数据结构分为（ ）两大类。

A．动态结构、静态结构 B．顺序结构、链式结构

C．线性结构、非线性结构 D．初等结构、构造型结构

答案（ **C**  ）

2、若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用（ ）存储方式最节省时间。

A．顺序表 B．双链表 C．带头结点的双循环链表 D．单循环链表

答案（  **A**  ）

3、循环队列A[0..m-1]存放其元素值，用front和rear分别表示队头和队尾，则当前队列中的元素数是( )。

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

C. rear-front-1 D. rear-front

答案（ **A**  ）

4、串的长度是指（ ）

A．串中所含不同字母的个数 B．串中所含字符的个数

C．串中所含不同字符的个数 D．串中所含非空格字符的个数

答案（  **B** ）

5、设广义表L=（（a,b,c）），则L的长度和深度分别为（ ）。

A. 1和1 B. 1和3 C. 1和2 D. 2和3

答案（ **C** ）

6、二叉树的先序遍历和中序遍历如下： 先序遍历：EFHIGJK；中序遍历: HFIEJKG 。该二叉树根的右子树的根是：( )

A. E B. F C. G D. H

答案（ **C**  ）

7、深度为h的满m叉树的第k层有（ ）个结点。(1=<k=<h)

A．mk-1  B．mk-1 C．mh-1 D．mh-1

答案（ **A** ）

8、关键路径是事件结点网络中（ ）。

A．从源点到汇点的最长路径 B．从源点到汇点的最短路径

C．最长回路 D．最短回路

答案（ **A** ）

9、散列文件使用散列函数将记录的关键字值计算转化为记录的存放地址，因为散列函数是一对一的关系，则选择好的（ ）方法是散列文件的关键。

A. 散列函数 B. 除余法中的质数

C. 冲突处理 D. 散列函数和冲突处理

答案（ **D** ）

10、m阶B-树是一棵( )

A. m叉排序树 B. m叉平衡排序树

C. m-1叉平衡排序树 D. m+1叉平衡排序树

答案（ **B** ）

|  |  |
| --- | --- |
| **得分** |  |

**二、判断题（本大题共10小题，每小题1分，总计10分）**

**(判断对1个题给1分，判断错1个题不给分)**

1、数据元素是数据的最小单位。( **×** )

2、对任何数据结构链式存储结构一定优于顺序存储结构。( **×** )

3、队列和栈都是运算受限的线性表，只允许在表的两端进行运算。（ **×** ）

4、数组不适合作为任何二叉树的存储结构。（ **×** ）

5、哈夫曼树无左右子树之分。( **×** )

6、拓扑排序算法把一个无向图中的顶点排成一个有序序列。（ **×** ）

7、B-树的插入算法中，通过结点的向上“分裂”，代替了专门的平衡调整。（  **√**  ）

8、倒排文件与多重表文件的次关键字索引结构是不同的。（  **√** ）

9、快速排序和归并排序在最坏情况下的比较次数都是O(nlog2n)。（ **×** ）

10、稀疏矩阵压缩存储后，必会失去随机存取功能。（ **√** ）

|  |  |
| --- | --- |
| **得分** |  |

**三、填空（本大题共10小题，每小题1.5分，总计10分）**

**(填对1个题给1.5分，填错1个题不给分)**

1、下面程序段中带下划线的语句的执行次数的数量级是：（ **log2n** ）

i=1； WHILE（i<n） i=i\*2 ；

2、在双向链表结构中，若要求在p指针所指的结点之前插入指针为s 所指的结点，则需执行下列语句：

s->next=p； s->prior= p->prior；p->prior=s；( **s->prior->next** ) =s；

3、队列是限制插入只能在表的一端，而删除在表的另一端进行的线性表，其特点是（ **先进先出**  ）。

4、已知一棵度为3的树有2个度为1的结点，3个度为2的结点，4个度为3的结点，则该树有（ **12** ）个叶子结点。

5、在完全二叉树中，编号为i和j的两个结点处于同一层的条件是(  **⎣log2i⎦ = ⎣log2j⎦**  )。

6、在有向图的邻接矩阵表示中，计算第I个顶点入度的方法是（ **第I列非零元素个数** 或意思相同的描述 ）。

7、在顺序表（8,11,15,19,25,26,30,33,42,48,50）中，用二分（折半）法查找关键码值20，需做的关键码比较次数为（ **4** ）.

8、外排序的基本操作过程是（ **生成有序归并段（顺串）** ）和归并。

9、链接存储的特点是利用（ **指针** ）来表示数据元素之间的逻辑关系。

10、用一维数组设计栈，初态是栈空，top=0。现有输入序列是 a、b、c、d，经过 push、push、pop、push、pop、push操作后，输出序列是（ bc ），栈顶指针是（ **2** 或 **d** 都给分 ）。

|  |  |
| --- | --- |
| **得分** |  |

**四、应用题（本大题共6小题，每小题6分，总计36分）**

1、利用两个栈sl,s2模拟一个队列时，如何用栈的运算实现队列的插入，删除以及判队空运算。请简述这些运算的算法思想。**(本题6分)**

答：栈的特点是后进先出，队列的特点是先进先出。初始时设栈s1和栈s2均为空。

（1）用栈s1和s2模拟一个队列的输入：设s1和s2容量相等。分以下三种情况讨论：若s1未满，则元素入s1栈；若s1满，s2空，则将s1全部元素退栈，再压栈入s2，之后元素入s1栈；若s1满，s2不空（已有出队列元素），则不能入队。

（2）用栈s1和s2模拟队列出队（删除）：若栈s2不空，退栈，即是队列的出队；若s2为空且s1不空，则将s1栈中全部元素退栈，并依次压入s2中，s2栈顶元素退栈，这就是相当于队列的出队。若栈s1为空并且s2也为空，队列空，不能出队。

（3）判队空 若栈s1为空并且s2也为空，才是队列空。

**（要点答出或意思说清楚就给分，若要点没有完全答出则酌情扣分）**

1. 已知一棵二叉树的后序遍历序列为EICBGAHDF，同时知道该二叉树的中序遍历序列为CEIFGBADH，试画出该二叉树。

|  |
| --- |
|  |

|  |
| --- |
| F |

|  |
| --- |
| E |

|  |
| --- |
| C |

|  |
| --- |
| I |

|  |
| --- |
| A |

|  |
| --- |
| B |

|  |
| --- |
| G |

|  |
| --- |
| D |

|  |
| --- |
| H |

解：**（共6分）（画对给6分，若有错误酌情扣分）**

1. 设有正文AADBAACACCDACACAAD,字符集为A,B,C,D,设计一套二进制编码，使得上述正文的编码最短。

解：**（共6分）（画对给5分，若有错误酌情扣分）**

|  |
| --- |
| 3 |

|  |
| --- |
| 5 |

|  |
| --- |
| 9 |

|  |
| --- |
| 0 |

|  |
| --- |
| 0 |

|  |
| --- |
| 0 |

|  |
| --- |
| 1 |

|  |
| --- |
| 1 |

|  |
| --- |
| 1 |

|  |
| --- |
|  |

（1）写出：字符A，B，C，D出现的次数为9,1,5,3

或相应意思则这**给1分。**

（2）画出哈夫曼树**给4分**。**（不一定必须与右图相同，**

**只要画正确就给分）**

（3写出其编码：

哈夫曼编码如下A:1

B:000

C:01

D:001

**（**哈夫曼编码不一定与上述一致，只要写对就给分）**给1分。**

4、已知某图的邻接表为如右图所示：

（1）写出此邻接表对应的邻接矩阵；

（2）写出由v1开始的深度优先遍历的序列；

（3）写出由v1开始的深度优先的生成树；

（4）写出由v1开始的广度优先遍历的序列；

（5）写出由v1开始的广度优先的生成树；

解：**（共6分）**

（1）**（2分）**

****

（2）V1V2V3V4V5 **（1分）** (4) V1V2V3V4V5**（1分）**

(3)**（1分）** **（1分）**

(5)

5、设哈希(Hash)表的地址范围为0～17，哈希函数为：H(K)=K MOD 16,K为关键字，用线性探测再散列法处理冲突，输入关键字序列:(10,24,32,17,31,30,46,47,40,63,49)造出哈希表，试回答下列问题：

(1)画出哈希表示意图；(2)若查找关键字63，需要依次与哪些关键字比较?

(3)若查找关键字60，需要依次与哪些关键字比较?

(4)假定每个关键字的查找概率相等，求查找成功时的平均查找长度。

解：**（共6分）**

（1）（3分）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 关键字 | 32 | 17 | 63 | 49 |  |  |  |  | 24 | 40 | 10 |  |  |  | 30 | 31 | 46 | 47 |
| 比较次数 | 1 | 1 | 6 | 3 |  |  |  |  | 1 | 2 | 1 |  |  |  | 1 | 1 | 3 | 3 |

（2）查找关键字63，H（k）=63 MOD 16=15，依次与31，46，47，32，17，63比较。（1分）

（3）查找关键字60，H（k）=60 MOD 16=12，散列地址12内为空，查找失败。（1分）

（4）ASLsucc=23/11（1分）

6、有一随机数组(25,84,21,46,13,27,68,35,20),现采用某种方法对它们进行排序,其每趟排序结果如下, 则该排序方法是什么?

初 始:25,84,21,46,13,27,68,35,20

第一趟:20,13,21,25,46,27,68,35,84

第二趟:13,20,21,25,35,27,46,68,84

第三趟:13,20,21,25,27,35,46,68,84

**答：该排序方法为快速排序。**

|  |  |
| --- | --- |
| **得分** |  |

**五、算法设计题（本大题共3小题，每小题8分，总计24分）**

1、已知一个单链表中每个结点存放一个整数，并且结点数不少于2，请设计算法以判断该链表中第二项起的每个元素值是否等于其序号的平方减去其前驱的值，若满足则返回ture，否则返回false.（8分）

解：**（共8分）**

typedef false 0

typedef true 1

typedef struct node

{

ElemType data;

struct node \* next;

}node,\*LinkList;

**int**  Judge（LinkedList la）

{ p=la->next->next；

pre=la->next；

i=2；

while（p!=null）

if（p->data==i\*i-pre->data）

{i++；

pre=p；

p=p->next；

}

else

break；

if（p!=null）

return（false）；

else

return（true）；

}

2、设一棵二叉树中各结点的值互不相同，其前序序列和中序序列分别存于两个一维数组pre[1..n ]和mid[1..n ]中，试遍写算法建立该二叉树的二叉链表。

解：**（共8分）**

typedef struct BiNode{

ElemType data;

Struct BiNode \*lchild , \* rchild;

}BiNode, \*BiTree;

void PreInCreat(BiTree root,ElemType pre[],ElemType in[],int l1,inth1,int l2,int h2)

{

root=(BiTree)malloc(sizeof(BiNode));

root->data=pre[l1];

for(i=l2;I<=h2;I++)

if(in[i]==pre[l1]

break;

if(i==l2)

root->lchild=null;

else

PreInCreat(root->lchild,pre,in,l1+1,l1+(i-l2),l2,i-1);

if(i==h2)

root->rchild=null;

else

PreInCreat(root->rchild,pre,in,l1+(i-l2)+1,h1,i+1,h2);

}

3、以邻接表为存储结构，写出图的深度优先搜索DFS算法的非递归算法。

解：**（共8分）**

**#**define VexNum 10

typedef struct node

{

int adjvex;

struct node \* nextarc;

}ArcNode;

typedef struct

{

vertype data;

ArcNode \* firstarc;

} Adjlist[VexNum];

typedef struct{

Adjlist adjlist;

int vexnum,arcnum;

}Graph;

Void DFS(Graph g, vertype v)

{

ArcNode \*stack[];

int top;

printf(v);

top=0;

p=g[v].firstarc;

while(top>0 || p!=NULL)

{

while(p)

if(p && visited[p->adjvex]) p=p->next;

else

{

printf(p->adjvex);visited[p->adjvex]=1;

stack[++top]=p;

p=g[p->adjvex].firstarc;

}

if(top>0)

{

p=stack[top--];

p=p->next;

}

}

}