**西南交通大学2019~2020学年第1学期期末考试A**

**班 级** **学 号** **姓 名**

**密封装订线**  **密封装订线**  **密封装订线**

**课程代码 3271035 课程名称 数据结构B 考试时间 120分钟**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **题号** | **一** | **二** | **三** | **四** | **五** | **六** | **总成绩** |
| **得分** |  |  |  |  |  |  |  |

**阅卷人签字：**

**注意：请将各题答案写在答题卷上。**

**一、单项选择题（每小题2分，共30分）**

1. **算法分析的目的是（ ），算法分析的两个主要方面是空间复杂度和时间复杂度**

**A. 找出数据结构的合理性**

**B. 研究算法中的输入和输出的关系**

**C. 分析算法的效率以求改进**

**D. 分析算法的易懂性和文档性**

1. **带头结点的单链表head为空的判定条件（ ）。**

**A. head==NULL B. head->next ==NULL**

**C. head->next == head D. head!=NULL**

1. **二维数组A中，每个元素长度为3个字节，行下标i从1到8，列下标j从1到10，从首地址SA开始连续存放在存储器内，该数组按行存放，则元素A[8][5]的起始地址是（ ）。**

**A. SA+141 B. SA+144 C. SA+220 D. SA+222**

1. **采用稀疏矩阵的三元组表形式进行压缩存储，若要完成对三元组表进行转置，只要将行和列对换，这种说法（ ）。**

**A. 正确 B. 错误 C. 无法确定 D. 以上均不对**

1. **假定在一棵二叉树中，度为2的结点数为16，度为1的结点数为25，则叶子结点数为（ ）个。**

**A. 17 B. 18 C. 19 D. 41**

1. **设结点x和y是二叉树中的任意2个结点，若在先根序列中x在y之前，而在后根序列中x在y之后，则x和y的关系是（ ）。**

**A. x是y的左兄弟 B. x是y的右兄弟**

**C. x是y的祖先 D. x是y的后代**

1. **设某数据结构的二元组形式表示为A=(D，R)，D={01，02，03，04，05，06，07，08，09}，R={r}，r={<01，02>，<01，03>，<01，04>，<02，05>，<02，06>，<03，07>，<03，08>，<03，09>}，则数据结构A是（ ）。**

**A. 线性结构 B. 树型结构 C. 物理结构 D. 图型结构**

1. **在一棵深度为h的具有n个元素的二叉排序树中，查找所有元素的最长查找长度为（ ）。**

**A. n B. log2n C. (h+1)/2 D. h**

1. **设图的邻接矩阵为，则该图为（ ）。**

**A. 有向图 B. 无向图 C. 强连通图 D. 完全图**

1. **设指针变量p指向单链表中结点A，若删除单链表中结点A，则需要修改指针的操作序列为（ ）。**

**A. q=p->next；p->next=q->next；free(q)；**

**B. q=p->next；p->data=q->data；free(p)；**

**C. q=p->next；p->data=q->data；p->next=q->next；free(q)；**

**D. q=p->next；q->data=p->data；p->next=q->next；free(q)；**

1. **已知表长为25的哈希表，用除留取余法，按公式H(key)=key MOD p 建立哈希表，则p应取（ ）为宜。**

**A. 23 B. 24 C. 25 D. 26**

1. **在对查找表的查找过程中，若被查找的数据元素不存在，则把该数据元素插到集合中，这种方式主要适合于（ ）。**

**A. 静态查找表　 B. 动态查找表**

**C. 静态查找表和动态查找表 D. 两种表都不适合**

1. **一组记录的排序码为(25,48,16,35,79,82,23,40,36,72)，其中含有5个长度为2的有序表，按归并排序算法对该序列进行一趟归并后的结果是（ ）。**

**A. 16 25 48 35 79 82 23 36 40 72**

**B. 16 25 35 48 79 82 23 36 40 72**

**C. 16 25 35 48 23 40 79 82 36 72**

**D. 16 25 35 48 79 23 36 40 72 82**

1. **树的基本遍历策略可分为先根遍历和后根遍历；二叉树的基本遍历策略可分为先序遍历、中序遍历和后序遍历。这里，我们把由树转化得到的二叉树叫做这颗树对应的二叉树。结论（ ）是正确的。**

**A. 树的先根遍历序列与其对应的二叉树的先序遍历序列相同**

**B. 树的后根遍历序列与其对应的二叉树的后序遍历序列相同**

**C. 树的先根遍历序列与其对应的二叉树的中序遍历序列相同**

**D. 以上都不对**

1. **判定循环队列QU（最多元素MaxSize，少用一个元素空间）为满的条件是（ ）。**

**A. QU->front==QU->rear**

**B. QU->front!=QU->rear**

**C. QU->front==(QU->rear+1)%MaxSize**

**D. QU->front!=(QU->rear+1)%MaxSize**

**二、判断题（正确的划“√”，错误的划“×”，每小题1分，共11分）**

1. **邻接表可以表示有向图，也可以表示无向图。（ ）**
2. **不论线性表采用顺序存储结构还是链式存储结构，删除值为X的结点的时间复杂度均为O(n)。（ ）**
3. **串的存储密度与结点大小有关。（ ）**
4. **二维数组和多维数组均是特殊的线性结构。（ ）**
5. **通常，二叉树的第i层上有2i-1个结点。（ ）**
6. **设一棵二叉树的先序序列和后序序列，则能够唯一确定出该二叉树的形状。（ ）**
7. **希尔排序算法的时间复杂度为O(n2)。（ ）**
8. **中序遍历一棵二叉排序树可以得到一个有序的序列。（ ）**
9. **如果两个关键字的值不等但哈希函数值相等，则称这两个关键字为同义词。（ ）**
10. **Prim算法的时间主要取决于边数，因此，它比较适合于稀疏图。（ ）**
11. **栈和队列的存储方式，既可以是顺序方式，又可以是链式方式。（ ）**

**三、填空题（每空1分，共18分）**

**1、在串S=“structure”中，以t为首字符的子串有 (1) 个。**

**2、假设一个9阶的下三角矩阵A按列优先顺序压缩存储在一维数组B中，其中B[0]存储矩阵中第1个元素a11，则a62存放的位置是 (2) 。**

**3、在快速排序、堆排序、归并排序中， (3) 排序是稳定的。**

**4、散列表中解决冲突的两种方法是 (4) 和 (5) 。**

**5、设某棵二叉树的中序遍历序列为ECADFB，后序遍历序列为ECFDBA，则其前序遍历序列为 (6) 。**

**6、中缀表达式6/y+3\*(x+2)所对应的后缀表达式为 (7) 。**

**7、请指出下述算法实现的功能 (8) 。**

**proc Demo(var s2: seqstack; s1:seqstack)**

**{ //seqstack是顺序栈类型**

**DataType tmp;**

**Initstack(tmp);**

**Initstack(s2);**

**while(!stackEmpty(s1))**

**{ //栈s1非空**

**x=pop(s1);**

**push(tmp,x);**

**}**

**while(!stackEmpty(tmp))**

**{x=pop(tmp); push(s1,x); push(s2,x)}**

**}**

**8、请指出下述算法功能 (9) 。**

**void function(Bitree \*t){**

**if(p != NULL){**

**function(p -> lchild);**

**function(p -> rchild);**

**printf(“%d”,p -> data);**

**}**

**}**

**8、在非空双向链表中，向q所指链结点前面插入一个由p所指链结点的操作对应的语句依次为：p->next=q; p->prior=q->prior; p->prior->next=p; (10) 。（空白处为一条赋值语句）**

**10、若广义表A=(a,b,(c,d),(e,(f,g)))，则式子**

**head(tail(head(tail(tail(A)))))的值是 (11) 。**

**11、若某完全二叉树的结点个数为100，则第60个结点的度为 (12) 。**

**12、判断一个无向图是不是一棵树的条件是 (13) 。**

**13、n个顶点的连通图至少有 (14) 边。**

**14、8个元素组成的有序顺序表（a0，a1，a2，a3，a4，a5，a6，a7）采用折半查找，经过2次关键字比较后，可以找到的元素有 (15) 。**

**15、在分块查找方法中，首先查找 (16) ，然后再查找相应的块表。**

**16、设待排序关键字存储为(12,27,8,77,3,65,16,97,5,70)，选择12作为枢轴，按非递减有序，经过一趟快速排序后结果为 (17) \_ 。**

**18、一个算法具有5个特性：有穷性， (18) ，可行性，有零个或多个输入，有1个或多个输出。**

**四、算法填空（每空2分，共10分）**

**1、下面求2个集合A和B的并集（A U B）的算法，集合A和B分别用单链表La和Lb的带头结点的单链表表示（链表中元素按升序排列），其中并集用单链表Lc表示（带头结点，其中数据按升序排列）**

**typedef struct ListNode{**

**int data ;**

**Struct ListNode \*next;**

**} ListNode,\*Linklist;**

**void Mergelist\_L(Linklist La, Linklist Lb, Linklist \*Lc)**

**{ pa= La->next; pb= Lb->next;**

**\*Lc = pc = ( ListNode \*)malloc(sizeof(ListNode));**

**while(pa && pb)**

**{ p=( ListNode \*)malloc(sizeof(ListNode));**

**if(pa->data<=pb->data)**

**{ p->data= ① ;**

**pc->next=p;**

**pc=p;**

**pa=pa->next;**

**}else**

**{ p->data= pb->data;**

**pc->next=p;**

**pc=p;**

**pb=pb->next;**

**}**

**}**

**pa=pa?pa:pb;**

**while( ② )**

**{ p=( ListNode \*)malloc(sizeof(ListNode));**

**p->data= pa->data;**

**pc->next=p;**

**pc=p;**

**pa=pa->next;**

**}**

**pc->next=NULL;**

**}**

**2、以下程序是求二叉树深度的递归算法，请填空将算法补充完整。**

**typedef struct bt\_node{**

**int data;**

**struct bt\_node \*left,\*right;**

**}BTNode;**

**int depth(BTNode \*bt){**

**int hl , hr;**

**if(bt == NULL) return ③ ;**

**hl=depth(bt->left);**

**hr=depth(bt->right);**

**if( ④ )**

**⑤ ;**

**return**

**hr+1;**

**}**

**五、综合题（共31分）**

**1、关键码序列(503，087，512，061，908，170，897，275，653，426)，执行希尔排序，增量数组dlta[]=｛5，3，2，1｝，请写出每一趟排序结束时的关键码状态。(4分)**

**2、使用克鲁斯卡尔算法构造如图1所示的一棵最小生成树，并给出构造过程。(6分)**

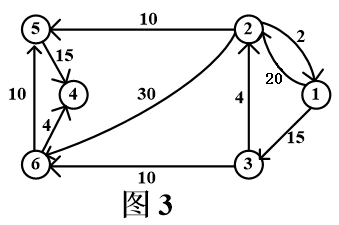


**3、****对关键字序列（72，87，61，23，94，16，05，58）进行堆排序，使之按关键字递减次序排列（小顶堆），请给出排序过程中得到的初始堆和前三趟的序列状态。(5分)**

**4、请将如图所示的森林转换成二叉树。（要求写出转换过程）(5分)**



**5、对如图3所示的有向图，试图利用Dijkstra算法求源点1到其他各顶点的最短路径，要求给出相应的求解步骤。(6分)**



**6、设电文由6个字符A，B，C，D，E，F组成，它们在电文中出现的次数分别为：10，4，8，3，2，7，画出对应的哈夫曼树，并写出每个字符的编码。（5分）**

**答题卷**

**一、单项选择题（每小题2分，共30分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1~5** |  | **6~10** |  | **11~15** |  |

**二、判断题（正确的划“√”，错误的划“×”，每小题1分，共11分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1~5** |  | **6~10** |  | **11** |  |

**三、填空题（每空1分，共18分）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** |  | **2** |  | | | **3** |  |
| **4** |  | **5** |  | | | **6** |  |
| **7** |  | **8** |  | | | | |
| **9** |  | | | **10** |  | | |
| **11** |  | **12** |  | | | **13** |  |
| **14** |  | **15** |  | | | **16** |  |
| **17** |  | | | | | **18** |  |

**四、算法填空（每空2分，共10分）**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** |  | **2** |  |
| **3** |  | **4** |  |
| **5** |  | | |

**五、综合题（共31分）**