私	十目代码: <u>925</u> 科目名称: <u>数据结构</u> 共 7 页, 第 1 页					
进	意事项:答案一律写在答题纸上,写在试卷上的不予装订和评分!					
1-	、填空题(本大题共 15 小题,每小题 2 分,共 30 分)					
lì.	栈中的数据元素是先进出,队列中的数据元素是先进出。					
2.	一个循环队列存于长度为7的一维数组 B[0,n-1]中,约定队列满时,队列中有6个元素。如果队					
	首指针 front 为指向队首元素的数组下标, 元素按顺时针方向入队,队尾指针 rear 为指向队尾元					
1	素在队列中顺时针方向的下一位置的数组下标,现在 front 和 rear 分别指向 5 和 3,则队列中元					
	素个数为 。					
3.	已知模式串 T='abcaababcabcab',则它的 next 函数值为。					
4.	对森林进行先序遍历,相当于对森林中的每一棵树从左到右进行遍历。					
5.	在结点个数为 n(n>1)的各棵树中,深度最大的树有层。					
6.	若二叉树先序遍历的扩展序列为 AB*D*EC**F***,其中*代表空链域,则二叉树的后序遍历序列					
	为					
7.	在 n 个顶点的有向连通图中最多有条弧。					
8.	用邻接矩阵来表示无向图,如果该图有9个顶点和8条边,则该邻接矩阵中有个非零元。					
9.	在赫夫曼树中,如果叶子结点数为 1000,则结点总数为。					
10.	已知一无向图 G=(V, E), 其中 顶点 V={a,b,c,d,e}, 边 E={(a,b),(a,d),(a,c),(d,c),(b,					
}	e)},现用某一种图的遍历方法从顶点 a 开始遍历图,得到的序列为 abecd,则采用的是					
	优先遍历方法。					
11.	一棵 m 阶 B_树的非终端结点至少有					
12.	对有 14 个关键字的有序表进行折半查找,在等概率的情况下查找成功时的平均查找长度为					
13.	深度为 4 的平衡二叉树的结点数至少有个。					
14.	对整型关键字序列(21,15,32,67,43,9,55,10)进行一趟快速排序得到的关键字序列是。					
15.	在最坏情况下,堆排序的时间性能为 O ()。					
二、	单选题(本大题共15小题,每小题2分,共30分)					
1.链表不具备的特点是 ()。						
	A. 可以随机访问任一结点 B. 插入删除不需要移动元素					
	C. 不必事先估计存储空间 D. 所需空间与其长度成正比					
2.已约	如不带头结点的双向循环链表中的结点结构为 (data,last,next), 在指针 p 所指结点之后插入由指					

注意事项,答案—律写在答题纸上,写在试卷上的不予装订和评分! 针。指向的新结点,相应操作为()。 A. p>next=s; s→last=p; p>next→last=s; s→next=p>next B. p>next=s; p>next=p>next; p>next=s; p>next=p>next C. s→last=p; s>next=p>next; p>next=s; p>next=s D. s>last=p; s>next=p>next; p>next=s; p>next=s 3.已知核 S 允许在两端出栈,但只允许在一端入栈; 队列 Q 只允许在一端入队列,在另一端出队列。 设核 S 和队列 Q 的功始状态为空。 e1,e2,e3, e4,e5,e6 依次通过栈 S,一个元素出栈后即进队列 Q,更不可能得到的出队列的顺序是()。 A. e2,e4,e3,e5,e1,e6 C. e5,e1,e6,e3,e2,e4 4厂文表 E=(a, (b),E)的长度和深度是()。 A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 S.设有 999 个无序的元素,希望用最快速度振出其中前 5 个最大的元素,采用下述哪种方法最好()。 A. 快速排序 B.推排序 C.基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反。则该二叉树。 定是 ()。 A. 空或只有一个结点	;	科目代码	马: <u>92</u>	5_ 科	目名称:	数据约	<u> </u>		井	7页,	第2页
A. p->next=s; s->last=p; p->next->last=s; s->next=p->next B. p->next=s; p->next->last=s; s->last=p; s->next B. p->next=s; p->next->last=s; s->last=p; s->next C. s->last=p; s->next=p->next; p->next->last=s D. s->last=p; s->next=p->next C. s->last=p; s->next=p->next D. s->last=p; s->next=p->next C. s->last=p; s->next=p->next D. s->last=p; p->next->last=s D. s->next=p->next D. shall sp C. sabtle sp p->next->last=s D. shall sp p->next=s D. s						,写在	试卷上的	不予装订和资	平分!		
B. p->next=s; p->next=>last=p; s->last=p; s->next=p>next C. s->last=p; s->next=p>next; p->next=s; p->next=s D. s->last=p; s->next=p->next; p->next=s; p->next=s 3.已知核 S. 允许在两端出栈,但只允许在一端入栈,队列 Q. 兄允许在一端入队列,在另一端出队列,设栈 S. 和队列 Q. 的初始状态为空。el,e2,e3, e4,e5,e6 依次通过核 S. 一个元素出栈后即进队列 Q. 贝不可能得到的出队列的顺序是()。 A. c2,e4,e3,e5,e1,e6 C. c5,e1,e6,e3,e2,e4 D. e4,e1,e3,e5,e2,e6 4.广义表 B=(a, (b), E)的长度和深度是()。 A. 3,2 B. 3,	1	ts 指向的	的新结点,	相应操作	为()	•					
C. s→last=p; s→next=p→next; p→next→last=s D. s→last=p; s→next=p→next; p→next→last=s; p→next=s 3.已知栈 S 允许在两端出栈,但只允许在一端入栈,队列 Q 兄允许在一端入队列,在另一端出队列,设栈 S 和队列 Q 的初始状态为空,e1,e2,e3, e4,e5,e6 依次通过栈 S,一个元素出栈后即进队列 Q,则不可能得到的出队列的顺序是()。 A. e2,e4,e3,e5,e1,e6 C. e5,e1,e6,e3,e2,e4 4.广义表 E=(a, (b), E)的长度和深度是()。 A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 5.设有 999 个无序的元素,希望用最快速度抵出其中前 5 个最大的元素,采用下述哪种方法最好 ()。 A. 快速排序 B. 堆排序 C. 基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反。则该二叉科一定是()。 A. 空或只有一个结点 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其()。 A. 右子树中最左下的结点 B. 右子树中最右下的结点	l	A.	p->next=s	; s->last=p;	p->next->	last=s; s	->next=p->n	ext			
D. s->last=p; s->next=p->next; p->next=s; p->next=s 3.已知栈 S 允许在两端出栈,但只允许在一端入栈;队列 Q 只允许在一端入队列,在另一端出队列。设栈 S 和队列 Q 的初始状态为空,el,e2,e3, e4,e5,e6 依次通过栈 S,一个元素出栈后即进队列 Q,则不可能得到的出队列的顺序是()。 A. e2,e4,e3,e5,e1,e6	l	В. 1	->next=s	; p->next->l	ast=s; s->	ast=p; s-	>next=p->n	ext			
3.已知栈 S 允许在两端出栈,但只允许在一端入栈,队列 Q 只允许在一端入队列,在另一端出队列。股栈 S 和队列 Q 的初始状态为空,el,e2,e3, e4,e5,e6 依次通过栈 S,一个元素出栈后即进队列 Q,则不可能得到的出队列的顺序是()。 A. c2,e4,e3,e5,e1,e6	1	C. :	;->last - p;	s->next=p->	>next; p->	next=s; ¡	o->next->las	t=s			
设栈 S 和队列 Q 的初始状态为空,e1,e2,e3, e4,e5,e6 依次通过栈 S,一个元素出栈后即进队列 Q,则不可能得到的出队列的顺序是()。 A. e2,e4,e3,e5,e1,e6			•		•						
不可能得到的出队列的顺序是()。 A. e2,e4,e3,e5,e1,e6	3	.已知栈 9	允许在西	丙端出栈, (旦只允许在	E一端入	.栈;队列Q	只允许在一端	入队列,	在另一	端出队列。
A. e2,e4,e3,e5,e1,e6 C. e5,e1,e6,e3,e2,e4 4.广义表 E=(a, (b),E)的长度和深度是 ()。 A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 5.设有 999 个无序的元素,希望用最快速度抵出其中前 5 个最大的元素,采用下述哪种方法最好 ()。 A. 件速排序 B. 推排序 C. 基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树、定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 ()。 A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线素二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 (**)。 A. 右子树中最左下的结点 B. 右子树中最右下的结点	18	₹栈 S 和I	从列Q的	初始状态为	空,el,e2	,e3, e4,e	5,e6 依次通	过栈S,一个	元素出栈	后即进队	人列 Q,则
C. e5,e1,e6,e3,e2,e4 4.广义表 E=(a, (b),E)的长度和深度是 ()。 A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 5.设有 999 个无序的元素,希望用最快速度抵出其中前 5 个最大的元素,采用下述哪种方法最好 ()。 A. 快速排序 B. 堆排序 C. 基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树一定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 ()。 A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线素二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 (***)。 A.右子树中最左下的结点 B.右子树中最右下的结点	7	可能得到	的出队	列的顺序是	().						
4.广义表 E=(a, (b),E)的长度和深度是()。 A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 5.设有 999 个无序的元素,希望用最快速度抵出其中前 5 个最大的元素,采用下述哪种方法最好()。 A. 快速排序 B. 堆排序 C. 基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树一定是()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线素二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(**)。 A.右子树中最左下的结点 B.右子树中最右下的结点		Α. ο	2,e4,e3,e	5,e1, e 6		В.	e2,e5,e1,e3,	e4,e6			
A. 3,2 B. 3, ∞ C. 不能确定 D. 以上都不对 5.设有 999 个无序的元素,希望用最快速度抵出其中前 5 个最大的元素,采用下述哪种方法最好 ()。 A. 快速排序 B. 堆排序 C. 基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树、定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 (A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 ()。 A.右子树中最左下的结点 B.右子树中最右下的结点		C. 6	5,e1,e6,e	3,e2 ,e4		D	. e4,e1,e3,e5	,e2,e6			
5.设有 999 个无序的元素,希望用最快速度挑出其中前 5 个最大的元素,采用下述哪种方法最好()。 A.快速排序 B.堆排序 C.基数排序 D.Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树一定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 ()。 A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 (***)。 A.右子树中最左下的结点 B.右子树中最右下的结点	4.	广义表E	=(a, (b),E)的长度	美和深度 是	()	0)		
()。 A.快速排序 B.堆排序 C.基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反、则该二叉树、定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 (A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 (***)。 A.右子树中最左下的结点 B.右子树中最右下的结点		Α.	3,2	B. 3,	ω <u>,</u>	不能	角定。D	. 以上都不对	(00)		
A.快速排序 B.堆排序 C.基数排序 D. Shell 排序 6.某二叉树的先序序列和后序序列正好相反,则该二叉树一定是 ()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是 (A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其 (***)。 A.右子树中最左下的结点 B.右子树中最右下的结点	5.	设有 999	个无序	的元素,希	望用最快	速度挑	出其中前 5	个最大的元素	采用	下述哪种	方法最好
6.某二叉树的先序序列和后序序列正好相反,则该二叉树、定是()。 A. 空或只有一个结点 B. 高度等于其结点数 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其()。 A.右子树中最左下的结点 B.右子树中最右下的结点	().			W.	563		3/3/30.			
A. 空或只有一个结点 C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(*)。 A.右子树中最左下的结点 B.右子树中最右下的结点		A.快速	排序	B.堆排序	C.基	数排序	D. She	排序			
C. 任一结点无左孩子 D. 任一结点无右孩子 7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(***)。 A.右子树中最左下的结点 B.右子树中最右下的结点	6.	某二叉树	的先序序	列和后序的	列正好村	反,则	该二叉树一	定是()			
7.排序方法中,关键字比较的次数与记录的初始排列无关的是(A. Shell 排序 B. 快速排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(*)。 A.右子树中最左下的结点 B.右子树中最右下的结点		A. 空	戊 只有一⁄	个结点		B.	高度等于其	结点数			
A. Shell 排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(')。 A.右子树中最左下的结点 B.右子树中最右下的结点		C. 任-	-结点无	生孩子 🦪	Kan a	D.	任一结点无	右孩子			
A. Shell 排序 C. 直接插入排序 D. 简单选择排序 8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(')。 A.右子树中最左下的结点 B.右子树中最右下的结点	7.	排序方法	中,关键	字比较的数	次数与记录	的初始	排列无关的	是(
8. 下面关于哈希查找的说法中,正确的是()。 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(')。 A.右子树中最左下的结点 B.右子树中最右下的结点		A. She	排序		B. 快速	排序		4			
 A. 哈希函数构造的越复杂越好,因为这样随机性好,冲突小 B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(**)。 A.右子树中最左下的结点 B.右子树中最右下的结点 		C. 直接	美插入排 序	予	D. 简单	色选择排	序				
B. 除留余数法是所有哈希函数中最好的 C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(**)。 A.右子树中最左下的结点 B.右子树中最右下的结点	8.	下面关于	F哈希查:	戏的说法中	,正确的	是() .				
C. 不存在特别好与坏的哈希函数,要视情况而定 D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其()。 A.右子树中最左下的结点 B.右子树中最右下的结点		A. 哈希	函数构造	造的越复杂	越好,因	为这样队	· 恒机性好,A	中突小		•	
D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(**)。 A.右子树中最左下的结点 B.右子树中最右下的结点		B. 除質	余数法是	是所有哈希i	函数中最	好的					
D. 若采用开放定址法处理冲突,则删除元素只需直接将该元素从哈希表中删去即可 9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(**)。 A.右子树中最左下的结点 B.右子树中最右下的结点		C. 不存	在特别如	子与坏的哈	希函数,	要视情况	几而定				
9.在遍历中序线索二叉树时,某结点既有左子树又有右子树,那么它的前驱是其(')。 A.右子树中最左下的结点 B.右子树中最右下的结点		D. 若邪	2用开放5	2 址法处理》	中突,则!	刑除元素	5只需直接#	各该元素从哈希	6表中删	去即可	
A.右子树中最左下的结点 B.右子树中最右下的结点	9.7										j.
	,	100									•
C/工1-4/1-4/1-1-10/21 D/T-1-4/1-1-10/21 D/T-1-4/1-1-10/21 D/T-1-4/1-1-10/21 D/T-1-4/1-1-10/21 D/T-1-4/1-1-10/21							¥0	•			10
10.对某个无向图的邻接矩阵来说,下列哪种说法是正确的()。	10							()	7.		

获取 考研经验/复试资料/考研资讯 关注微信公众号 计算机与软件考研

科目代码: <u>925</u> 科目名称: <u>数据结构</u> 共 7 贝,第 3 贝 注意事项: 答案一律写在答题纸上,写在试卷上的不予装订和评分!	
A. 第 i 行上的非零元素个数和第 i 列的非零元素个数一定相等	
B. 矩阵中的非零元素个数等于图中的边数	
C. 第 i 行上和第 i 列上非零元素总数等于顶点 Vi 的度数	
D. 矩阵中非全零行的行数等于图中的顶点数	
11.图的广度优先搜索类似于二叉树的()遍历。	
A. 先根 B. 中根 C. 后根 D. 层次	
12.广义表 A=((x, (a,b)), (x,(a,b),y)), 则运算 Head(Head(Tail(A)))结果为 ()。	
A. x B. (a,b) C. (x, (a,b)) D. A	
13. 若一个栈以向量 V[1n]存储, 栈内有 s 个元素 (s <n), s+1,="" td="" top="" x="" 为="" 则下面="" 栈顶指针="" 进栈的<=""><td>ģ</td></n),>	ģ
正确操作是()。	
A. top:=top+1; V [top]:=x; top:=top+1	
C. top:=top-1; V [top]:=x; \ D. V [top]:=x; top:=top-1	
14. 以下序列不是堆的是《》)。	
A. (100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66)	
B. (100, 98, 85, 82, 80, 77, 66, 60, 40, 20, 10)	
C. (10, 20, 40, 60, 66, 77, 80, 82, 85, 98, 100)	
D. (100, 85, 40, 77, 80, 60, 66, 98, 82, 10, 20)	
15.一棵完全二叉树有 2015 个叶子结点,则结点总数为 ()。	
A. 4030 B. 4029 C. 4031 D. 4030 或 4029	
三、判断对错题(本大题共15小题,每小题1分,共15分)	
1. 数据结构中数据元素之间的逻辑关系称为数据的逻辑结构。()	
2. 在哈夫曼树中没有度为 1 的结点。 ()	
3. 若一个广义表的表头是空表,则该广义表一定是个空表。()	
4. 设有两个串 S1 和 S2, 求 S2 在 S1 中首次出现位置的运算称为求子串。()	
5. 有向图的邻接矩阵中,第 i 行上和第 i 列上非零元素总数等于顶点 Vi 的度数。()	
6. 具有 n 个顶点和 e 条边的无向图用邻接表存储表示时,需要 n 个头结点和 e 个边结点。()	
7. 所有内部排序方法都是基于关键字比较的排序方法。()	
3. 在非空的平衡二叉树中插入一个结点,原有结点中至少一个结点的平衡因子会改变。()	
D. 哈希表的平均查找长度与处理冲突的方法无关。()	
SV-IV DIVICED A CANAL IN DIVINOUS AND MINIMISE AND A MALE AND A MARKET AND A MINIMISE AND A MINI	

科目代码: __925_ 科目名称: 数据结构

共7页,第4页

注意事项: 答案一律写在答题纸上, 写在试卷上的不予装订和评分!

- 10. 在用堆排序算法排序时,如果要进行关键字降序排序,则需要建立"小顶堆"。()
- 11. 一棵完全二叉树的第6层上有9个叶子结点,则结点总数一定是40。()
- 12. 对 AOV 网进行拓扑排序时, 得到的拓扑有序序列不一定是唯一的。()
- 13. 如果一个有向图恰有一个顶点的入度为 0, 其余顶点的入度均为 1, 则是一棵有向树。()
- 14. 简单选择排序是一种不稳定的排序方法。()
- 15. 关键路径是 AOE 网中从源点到汇点的最短路径。(

四、综合题(本大题共2小题,每小题5分,共35分)

1. 如果树采用孩子兄弟链表表示法,树以及森林可以和二叉树相互转换,请把如图 4-1 所示的二叉树转换成相应的森林。

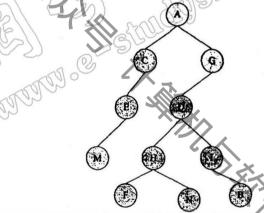


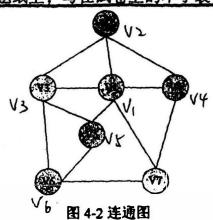
图 4-1 孩子兄弟链表表示的二叉树

- 2. 表长度为 13,数组的存储地址为 a[0] 至 a[12], 哈希函数为 H(key)=key MOD 13,使用链地址 法解决冲突,请画出对于给定的如下一组关键字{12,51,8,22,26,80,11,16,79,55}所得到 的哈希表,并求出等概率情况下查找成功和查找不成功时的平均查找长度。
- 3. 当三叉树中只有度为 3 和 0 的结点时,证明 $n_s=(n_0-1)/2$,其中 n_s 是度为 3 的结点数, n_0 是度为 0 的结点数。
- 4. 按顺序输入一组关键字序列(40,30,20,50,45,60),画出其构成的平衡二叉树,给出过程。
- 5. 对图 4-2 所示的连通图,写出深度优先搜索遍历序列和广度优先搜索遍历序列(从 V1 开始,下标小的邻接点优先),度为 3 的顶点有几个?
- 6. 已知一棵二叉树的层次遍历序列为 ABCDEFGHIJK 和中序遍历序列为 GKJDBEHACIF, 试画出该树。

科目代码: 925 科目名称: 数据结构

共7页,第5页

注意事项: 答案一律写在答题纸上, 写在试卷上的不予装订和评分!



7. 在图 4-3 的带权有向图中,求从顶点 VO 到其余各个顶点的最短路径,给出过程。

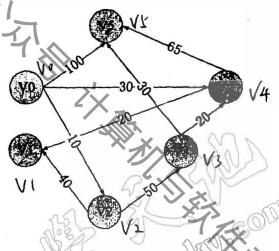


图 4-3 带权有向图

五、算法填空题(本大题共10小题,每小题2分,共20分)

1. 下列算法实现单循环链表的逆置。 Status Contray Cirl (LinkList &L) { /t 指向单循环链表的头结点 p=t->next; //p 指向单循环链表的第一个结点 q=p->next; //q 指向单循环链表的第二个结点 While (p!=L) { //让 p 结点 next 域指针指向其前驱 p->next=_ (1)//顺链向后移动指针 t t=p; //顺链向后移动指针 p p=q; (2) //顺链向后移动指针 q q=__

科目代码: 925 共7页、第6页 科目名称: 数据结构 注意事项:答案一律写在答题纸上,写在试卷上的不予装订和评分! (3) ; //让L的 next 域指针指向新链表的第一个结点 I ~>next= return OK: 2. 下列算法用于求二叉树的深度。 int Depth (BiTree T) { // 用 depthval 返回二叉树的深度 depthval = (4); if (!T) else depthLeft = Depth(T->Ichild); depthRight= Depth(T->rchild); depthval = (5) + (depthLeft > depthRight return depthval; 3.下列算法用于进行一趟希尔排序 void ShellInsert (SqList &L, int dk) { // 趙增量为 dk 的插入排序 for (i=dk+1; i<=n; ++i) if (L.r[i].key<_____(7) L.r[0] = (8)for (j=i-dk; j>0&&(L.r[0].key<L.r[j].key)THE TANK THE PROPERTY OF THE PARTY OF THE PA i=dkL.r[j+dk] = (9)L.r[j+dk] = (10)六、算法说明题(本大题共1题,共10分) 1、请说明下列算法中的 create 和 MS 的功能,图示 create 建立的结构,最后的运行结果是什么? #include<stdio.h> #include "malloc.h" #define n 5 static char ch[n]= $\{a',b',c',d',e'\}$; typedef struct node { int adjvex; struct node *next; } EdgeNode; typedef struct vnode { char vertex; EdgeNode *firstedge; } VertexNode[n]; int path[n], visited[n], sum=0; VertexNode G; void create() {int i,j,k; EdgeNode *s; { G[i].vertex=ch[i]; G[i].firstedge=0; } for(i=0; i< n; i++)

for(i=0; i<n; i++) { for(j=0; j< n; j++)

科目代码: 925 科目名称: 数据结构

共7页,第7页

注意事项:答案一律写在答题纸上,写在试卷上的不予装订和评分!

```
\{ if(a[i][i]=1) \}
        { s=(EdgeNode*)malloc(sizeof(EdgeNode));
                                        G[i].firstedge=s;
          s->adjvex=j; s->next=G[i].firstedge;
     }
   }
void MS(int i)
    { EdgeNode * p;
                              visited[i]=1
                                            p=G[i].firstedge;
             %c",G[i].vertex);
     printf("
     while(p)
      { if(visited[p->adjvex]==0) MS(p
                                                 计算机/软件工程专业
       p= p->next;
                                                     每个学校的
                                              考研真题/复试资料/考研经验
void MSTO
    { int k;
                                                考研资讯/报录比/分数线
     for(k=0;k<n;k++) visited[k]=0;
     for(k=0;k<n;k++)
                                                      免费分享
        if(visited[k]=0) MS(k);
                                                          微信 扫一扫
                                                         关注微信公众号
main()
{ create();
           MSTO; }
                                                        计算机与软件考研
七、算法设计题(本大题共1题,共10分)
1、稀疏矩阵用三元组顺序表来表示,数据结构定义如下:
                                         14
                                                0
                                     T=
                                                28
                                                0
  #define MAXSIZE 1000 //非零元个数的最大值
  typedef struct {
                //该非零元的行下标和列下标
      int i, j;
      ElemType e: // 该非零元的值
  } Triple; // 三元组类型
 typedef union {
      Triple data[MAXSIZE + 1]; //非零元的三元组表,data[0]未用
            mu, nu, tu; //矩阵的行数、列数和非零元个数
     int
 } TSMatrix; // 稀疏矩阵类型
   编写算法把稀疏矩阵 M 转置为 T (M 和 T 均为 TSMatrix 类型),要求算法的时间复杂度必须为
O(nu+tu), 并写出表示转置后的矩阵 T 的 data 数组的内容。
```