

齐鲁工业大学 21/22 学年第 一 学期《数据结构》
考试试卷（A 卷）答案及评分标准

一、术语解释题（本题满分 10 分）

1、逻辑结构和存储结构（3 分）

数据的逻辑结构是从**逻辑关系**上描述数据，-----1'

与数据的存储**无关**，是**独立**于计算机的。-----1'

数据的存储结构是数据对象在计算机中的**存储表示**。-----1'

2、栈和队列（3 分）

栈是**限定**仅在表尾进行插入或删除操作的线性表，-----1'

是**后进先出**的线性表。-----1'

队列是**先进先出**的线性表，只允许在表的一端进行插入，而在另一端删除元素。-----1'

3、平衡二叉树和平衡因子(4 分)

平衡二叉树或者是空树，或者是具有如下特征的二叉排序树；-----1'

左子树和右子树的深度之差的绝对值不超过 1；-----1'

左子树和右子树也是平衡二叉树。-----1'

平衡因子:该结点左子树与右子树的高度差。-----1'

二、简答计算题（本题满分 20 分，每小题 5 分）

1、设广义表 $L = (a, (b, c, d), e)$,

(1) 长度 3, -----1'

深度 2 -----1'

(2) Head (Tail (Tail (L))) 操作的结果: e。-----1'

(3) b 用 Head ()、 Tail () 从 L 中取出: Head (Head (Tail (L)))。-----2'

2、设有数组 $A[0 \cdots 7, 0 \cdots 9]$, 数组的每个元素长度为 3 字节, 数组从内存地址 100 开始顺序存放, 分别计算以行序为主序和以列序为主序存储元素 $A[5, 8]$ 的地址。

行序: 274 -----错了扣 2'

列序: 307 -----错了扣 2'

-----全错扣 5 分

3、循环队列 Q 存储在数组 $A[0..m-1]$ 中, 队头指针为 front, 队尾指针为 rear, 该循环队列采用少利用一个元素空间的方式判断队满和队空, 请回答:

- (1) 队满的条件? $(rear+1) \% m == front$ -----2'
 (2) 队空的条件? $front == rear$ -----1'
 (3) 如何求队列长度? $(rear-front+m) \% m$ -----2'

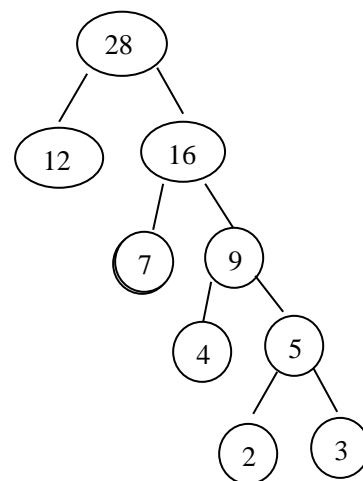
4、有 5 个元素，其入栈次序为：A，B，C，D，E，在各种可能的出栈次序中，以元素 C 和 D 最先出栈（即 C 第一个且 D 第二个出栈）的次序有哪几个？

CDEBA CDBEA CDBAE ----- 少一个扣 2 分，全错扣 5 分

三、分析应用题（本题满分 50 分，第 1-5 题每题 8 分，第 6 题 10 分）

1、已知下列字符 A、B、C、D、E 的权值分别为 3、12、7、4、2，请在表中写出其对应哈夫曼树 HT 存储结构的终态，并在表右侧画出这棵哈夫曼树。

	weight	parent	lch	rch
1	3	6	0	0
2	12	9	0	0
3	7	8	0	0
4	4	7	0	0
5	2	6	0	0
6	5	7	5	1
7	9	8	4	6
8	16	9	3	7
9	28	0	2	8

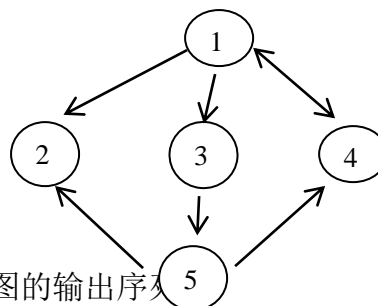


----表中每列占 1 分，

----哈夫曼树 4 分，错一处扣 1 分

2、已知有向图有 5 个顶点{1, 2, 3, 4, 5}，邻接表存储结构如下图所示：

(1) 请画出该图。



——图占 4 分，错一条边扣 1 分

(2) 从顶点 1 出发，按存储写出采用 DFS 遍历该图的输出序列

1 3 4 5 2 ——2 分，局部有错扣 1 分

(3) 从顶点 1 出发，按存储写出采用 BFS 遍历该图的输出序列。

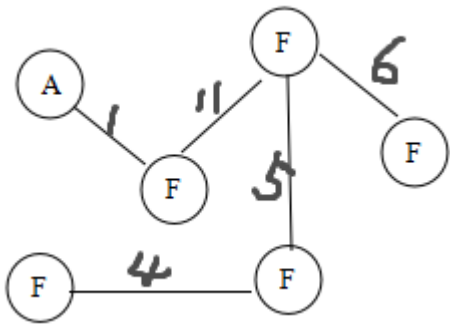
1 3 2 4 5 ——2 分，局部有错扣 1 分

3、有向网如右图所示，试用迪杰斯特拉算法求出从顶点 a 到其他各顶点间的最短路径，完成表 1。————每列 占 2 分

表 1

D 终点	i=1	i=2	i=3	i=4	i=5
c	<u>2</u> (a,c)				
d	12 (a,d)	12 (a,d)	11 (a,c,f,d)	<u>11</u> (a,c,f,d)	
e	∞	10 (a,c,e)	<u>10</u> (a,c,e)		
f	∞	<u>6</u> (a,c,f)			
g	∞	∞	16 (a,c,f,g)	16 (a,c,f,g)	<u>16</u> (a,c,f,g)
S 终点集	{a,c}	{a,c,f}	{a,c,f,e}	{a,c,f,e,d}	{a,c,f,e,d,g}

4、请根据 prim（普利姆）算法画出下图以顶点 A 为起始点的最小生成树的生成过程。



1, 11, 5, 4, 6 边选取的顺序
AF, FB, BD, DE, BC

-----错一处扣 1 分，扣满 8 分为止

5、构造哈希表，

0	1	2	3	4	5	6
21	12	14	24		4	
1	1	3	2		1	

$$ASL_{succ} = (1 \times 3 + 3 \times 2) / 5 = 8 / 5 = 1.6$$

$$ASL_{unsucc} = (5 + 4 + 3 + 2 + 1 + 2 + 1) / 7 = 18 / 7 = 2.57$$

6、设待排序的关键字序列为{5, 2, 7, 9, 4, 3, 8}, 写出以下排序方法的排序结果。

- (1) 直接插入排序 (前 2 趟); 2 5 7 9 4 3 8, 2 5 7 9 4 3 8 -----各 1 分
- (2) 冒泡排序 (前 2 趟); 2 5 7 4 3 8 9, 2 5 4 3 7 8 9 -----各 1 分
- (3) 简单选择排序 (前 2 趟); 2 5 7 9 4 3 8, 2 3 7 9 4 5 8 -----各 1 分
- (4) 2 路归并排序 (前 2 趟); 2 5 7 9 3 4 8, 2 5 7 9 3 4 8 -----各 1 分
- (5) 快速排序 (第 1 趟): 3 2 4 5 9 7 8 -----2 分

四、算法设计题 (本题满分 20 分, 每题 10 分)

1、将两个递增的带头结点的有序链表 La 和 Lb 合并为一个递增的有序链表 Lc。

```
void MergeList(LinkList &La, LinkList &Lb, LinkList &Lc)
```

```
{//合并链表 La 和 Lb, 合并后的新表使用头指针 Lc 指向
```

```
pa=La->next; pb=Lb->next;        -----初始化 1'
```

```
//pa 和 pb 分别是链表 La 和 Lb 的工作指针, 初始化为相应链表的第一个结点
```

```
Lc=pc=La; //用 La 的头结点作为 Lc 的头结点
```

```
while(pa && pb)                        -----循环 1'
```

```
{if(pa->data<pb->data) -----分支条件判断 1'
```

```
{pc->next=pa;pc=pa;pa=pa->next;} -----处理 1'
```

```
//取较小者 La 中的元素, 将 pa 链接在 pc 的后面, pa 指针后移
```

```
else if(pa->data>pb->data) -----分支条件判断 1'
```

```
{pc->next=pb; pc=pb; pb=pb->next;} -----处理 1'
```

```
//取较小者 Lb 中的元素, 将 pb 链接在 pc 的后面, pb 指针后移
```

```
else //相等时取 La 中的元素, 删除 Lb 中的元素 -----处理 2'
```

```
{pc->next=pa;pc=pa;pa=pa->next;
```

```
q=pb->next;
```

```
delete pb ;
```

```
pb =q;
```

```
}
```

```
}
```

```
pc->next=pa?pa:pb;    //插入剩余段    -----条件处理 1'
```

```
delete Lb;                //释放 Lb 的头结点    -----处理 1'
```

```
}
```

2、请设计非递归折半查找算法，若查找成功，返回 k 所在位置，若查找失败则返回 0。

```
int Search_Bin (SSTable ST, KeyType k){
    low=1;high=ST.length;          -----初始化 1'
    while(low<=high){              -----循环 1'
        mid=(low+high)/2;
        if(k==ST.R[mid].k)         -----分支条件判断 1'
            return mid;             -----处理 2'
        else if(key<ST.R[mid].k)   -----分支条件判断 1'
            high=mid-1; //前一子表查找 -----处理 1'
        else
            low=mid+1; //后一子表查找 -----处理 1'
    }
    return 0;                       //表中不存在待查元素 -----查找失败处理 2'
}
```