

一、 选择题（每题 2 分，共 10 分）

1. 数据结构是研究数据的_____以及对它们所进行的运算。
A. 抽象结构和理想结构 B. 理想结构和物理结构
C. 逻辑结构和物理结构 D. 抽象结构和物理结构
2. 对于顺序存储的线性表，设其长度为 n ，在任何位置上插入或删除操作都是等概率的。插入一个元素时所需移动元素次数的期望值为_____。
A. $n/2$ B. $(n+1)/2$ C. $(n-1)/2$ D. n
3. 若让元素 1, 2, 3, 4, 5 依次进栈，则出栈次序不可能出现在_____种情况。
A. 5, 4, 3, 2, 1 B. 2, 1, 5, 4, 3 C. 4, 3, 1, 2, 5 D. 2, 3, 5, 4, 1
4. 一个具有 1025 个结点的二叉树的高 h 为_____。
A. 10 B. 11 C. 10 至 1024 之间 D. 11 至 1025 之间
5. 若从无向图的任意一个顶点出发进行一次深度优先搜索可以访问图中所有的顶点，则该图一定是_____图。
A. 非连通 B. 连通 C. 强连通 D. 有向

二、 填空题（每题 2 分，共 10 分）

1. 设栈 S 和队列 Q 的初始状态为空，元素 e_1 、 e_2 、 e_3 、 e_4 、 e_5 和 e_6 依次进入栈 S ，一个元素出栈后即进入 Q ，若 6 个元素出队的序列是 e_2 、 e_4 、 e_3 、 e_6 、 e_5 和 e_1 ，则栈 S 的容量至少应该是_____。
2. 数组 $A[0..5, 0..6]$ 的每个元素占 5 个字节，将其按列优先次序存储在起始地址为 1000 的内存单元中，则元素 $A[5, 5]$ 的地址是_____。
3. 广义表 $(((a, b, c), d, e, f))$ 的长度是_____。
4. 表达式 $a*(b+c)-d$ 的后缀表达式是_____。
5. 一个具有 n 个顶点的无向图最多有_____条边。

三、 判断改错题（每题 2 分，共 10 分，对的打，错的打×并改正）

1. 树形结构和图形结构都是非线性结构，因此只能用链式方式来存储。（ ）
2. 设栈采用顺序存储结构。若已有 $i-1$ 个元素入栈，则将第 i 个元素入栈时，入栈算法的时间复杂性为 $O(1)$ 。（ ）
3. 字符串“aababaaaba”的 next 数组的值是-1 0 1 0 1 0 1 2 2 3。（ ）
4. 广义表的取表尾运算，其结果通常是个表，但有时也可是个单元素值。（ ）
5. 将一棵树转换成二叉树后，根结点没有左子树。（ ）

四、 算法阅读题（每题 5 分，共 15 分）

1. 已知顺序表 L 中的元素为整数，顺序表类型如下：

```
typedef struct {  
    int elem[MaxSize];  
    int last; //顺序表中最后一个元素的下标  
} SeqList;
```

(1) 若有一顺序表 L 中的元素为(20,15,17,18,9,6,8)，经过如下的 Proc1 运算处理之后，发生了什么变化？

(2) 总结 Proc1 运算的功能，并分析 Proc1 运算的时间复杂度。

```
void Proc1(Seqlist *L) {
    pos=0;
    for( i=0;i<=L->last;i++) {
        if(L->elem[i]%2!=0) {
            if(i!=pos) { temp=L->elem[i];L->elem[i]=L->elem[pos];L->elem[pos]=temp; }
            pos++;
        }
    }
}
```

2. 用带头结点的单链表表示一元多项式，链表中结点的类型如下：

```
typedef struct Polynode {
    int coef;    //系数域
    int exp;     //指数域
    struct Polynode *next; //指向下一个结点的指针域
} Polynode, * Polylist;
```

(1) 若有一元多项式 $A(x)=7+3x+2x^3+x^7$ ，请画出该多项式的存储结构示意图；

(2) 该一元多项式 A(x)经过如下的 Proc2 运算处理之后，发生了什么变化？

(3) 总结 Proc2 运算的功能。

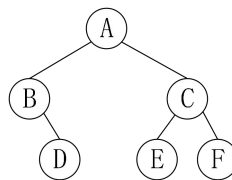
```
void Proc2(Polylist pa) {
    pre=pa;
    p=pa->next;
    while(p) {
        if(p->exp==0) { pre->next=p->next; free(p); }
        else { p->coef=p->coef*p->exp; p->exp--; pre=p; }
        p=pre->next;
    }
}
```

3. 二叉树采用二叉链表存储结构，结点有三个域，数据域 data，左指针域 lchild 和右指针域 rchild。

以下算法 Proc3 基于队列的基本运算设计实现。

二叉链表中的结点类型如下：

```
typedef struct node {
    ElemType data;
    struct node *lchild, *rchild;
} BiNode, *BiTree;
```



(1) 若有二叉树如上图所示，经过如下的 Proc3 运算处理之后，会得到什么结果？

(2) 总结 Proc3 运算的功能。

```
int Proc3(BiTree t){
    if(!t) return 0;
    InitQueue(&Q); /*初始化后：Q.front=Q.rear=0，Q.rear 指向队尾元素的下一个位置*/
    temp_w=0;
```

```

max_w=0;
last=-1;
EnQueue(&Q,t);
while(!QueueEmpty(&Q)){
    DeQueue(&Q,&p);
    temp_w++;
    if(p->lchild) EnQueue(&Q,p->lchild);
    if(p->rchild) EnQueue(&Q,p->rchild);
    if(Q.front>last){
        last=Q.rear-1;
        if(temp_w>max_w) max_w=temp_w;
        temp_w=0;
    }
}
return max_w;
}

```

五、 算法设计题（共 20 分）

1. （7 分）已知两个顺序表 LA 和 LB，其元素均为非递减有序排列，请设计算法 Merge(SeqList*LA, SeqList*LB, SeqList*LC)，将它们合并成一个顺序表 LC，要求 LC 也是非递减有序排列。

顺序表的类型如下：

```

typedef struct {
    ElemType elem[MaxSize];
    int last;
} SeqList;

```

2. （7 分）已知一个元素值递增有序的单链表 L（允许出现值重复的结点），设计算法 Remove(LinkList L)，删除 L 中值重复的结点，并分析算法的时间复杂度。

单链表中结点类型如下。

```

typedef struct Node {
    ElemType data;
    struct Node * next;
}Node,* LinkList;

```

3. （6 分）二叉树采用二叉链表存储结构，结点有三个域，数据域 data，左指针域 lchild 和右指针域 rchild。请设计算法 int NonLeafCount(BiTree t)，实现统计二叉树中非叶子结点的数目。二叉链表中的结点类型如下：

```

typedef struct node {
    ElemType data;
    struct node *lchild, *rchild;
} BiNode, *BiTree;

```

六、 综合应用题（共 35 分）

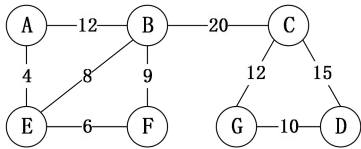
1. （5 分）画出如下所示的稀疏矩阵 A 的十字链表存储结构示意图。

$$A = \begin{bmatrix} 0 & 8 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 9 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 4}$$

2. （6 分）假设用于通信的电文仅由 5 个字母组成，字母在电文中出现的次数分别为 3，8，7，2，5（单位略）。

- （1）请为这 5 个字母设计哈夫曼编码。
- （2）请设计另一种由二进制表示的等长编码。
- （3）比较两种方案的优缺点。

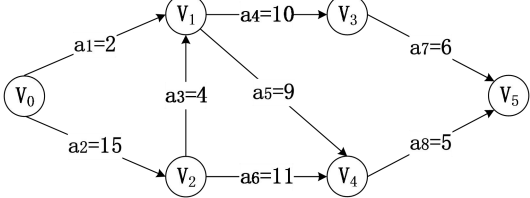
3. （7 分）为实现村村通工程，某公路局要在以下 7 个核心乡镇修建公路，下图为乡镇之间的距离示意图。



- （1）如果用邻接表来存储该图，请画出该图对应的邻接表存储结构示意图；
- （2）请给出一个方案，使得既满足村村通的要求，又使得修建公路的费用最节约。

4. （7 分）下图所示的 AOE 网表示一项工程，请解答下列问题：

- （1）每个事件的最早发生时间和最晚发生时间；
- （2）每个活动的最早开始时间和最晚开始时间；
- （3）完成此工程，至少需要多少时间？哪些活动是关键活动，并给出关键路径。



5. （5 分）设哈希函数 $H(\text{Key}) = \text{Key} \bmod 11$ ，哈希地址空间为 0~10，对关键字序列（32，13，49，24，38，21，4，12），按链地址法解决冲突的方法构造哈希表，并分别求出等概率下查找成功时和查找失败时的平均查找长度 ASL_{succ} 和 ASL_{unsucc} 。

6. （5 分）设待排序记录的关键字序列为：（46，55，13，42，94，17，05，70），请给出建立初始堆（小顶堆）的过程。

试卷答案(A 卷)

一、 单选题（每题 2 分，共 10 分）

1	2	3	4	5
C	A	C	D	B

二、 填空题（每题 2 分，共 10 分）

1	2	3	4	5
3	1175	1	abc+*d-	n(n-1)/2

三、 判断改错题（每题 2 分，共 10 分，对的打√，错的打×并改正）

1	×	也可以用顺序方式来存储。
2	√	
3	√	
4	×	广义表的取表尾运算，其结果一定是个表。
5	×	将一棵树转换成二叉树后，根结点没有右子树。

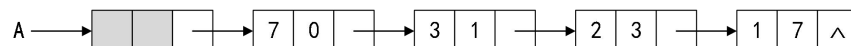
四、 算法阅读题（每题 5 分，共 15 分）

1.

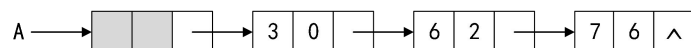
- (1) 顺序表 L 中的元素经过 Proc1 运算处理之后由原来的(20,15,17,18,9,6,8)变为(15,17,9,18,20,6,8);
- (2) Proc1 运算的功能：将顺序表奇数元素排列在前面，偶数元素排列在后面；
- (3) Proc1 运算的时间复杂度：O(n)。

2.

- (1) 一元多项式 $A(x)=7+3x+2x^3+x^7$ 的存储结构示意图如下：



- (2) 该一元多项式 $A(x)$ 经过 Proc2 运算处理之后，由原来的 $7+3x+2x^3+x^7$ 变为 $3+6x^2+7x^6$ ；或者其存储结构示意图如下：



- (3) Proc2 运算的功能：求一元多项式的一阶导数。

3.

- (1) 如图的二叉树 Proc3 运算处理之后，得到的结果为 3；
- (2) Proc3 运算的功能：计算二叉树的宽度。

五、 算法设计题（共 20 分） 参考答案，答案不唯一

1、（7 分）

```
void merge(SeqList*LA, SeqList*LB, SeqList*LC) {
    i=0; j=0; k=0;
    while(i<=LA->last&& j<=LB->last)
        if(LA->elem[i]<=LB->elem[j]) { LC->elem[k]=LA->elem[i]; i++; k++; }
        else { LC->elem[k]=LB->elem[j]; j++; k++; }
    }
    while(i<=LA->last) { LC->elem[k]= LA->elem[i]; i++; k++; }
    while(j<=LB->last) { LC->elem[k]= LB->elem[j]; j++; k++; }
    LC->last=LA->last+LB->last+1;
}
```

2、（7 分）

```
void Remove(LinkList L) {
    if(L->next==NULL) return;
    p=L->next;
    while(p->next) {
        if(p->next->data==p->data) { r=p->next; p->next=r->next; free(r); }
        else p=p->next;
    }
}
```

算法的时间复杂度为： $O(n)$

3、（6 分）

```
int NonLeafCount(BiTree t) {
    if(!T) return 0; /*空树没有结点*/
    else if(!T->lchild && !T->rchild)
        return 0; /*叶子结点*/
    else
        return (1 + NoLeafCount(T->lchild) + NoLeafCount(T->rchild));
}
```

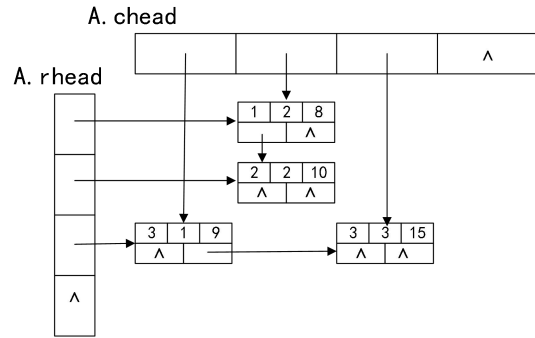
或

```
int NonLeafCount(BiTree t) {
    InitStack(&S);
    int nonleaf=0;
    while(t||!StackEmpty(&s)) {
        if(t&&(t->lchild|| t->rchild)) { push(&s,t); nonleaf++; t=t->lchild; }
        else { pop(&s,&p); t=p->rchild; }
    }
    return nonleaf;
}
```

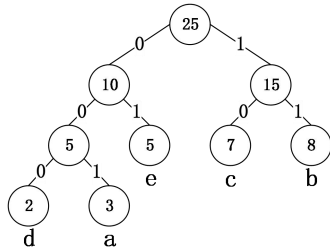
六、 综合应用题（共 35 分）

1、（5 分）

矩阵 $A = \begin{bmatrix} 0 & 8 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 9 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 4}$ 的十字链表存储:



2、（6 分）（1）假设这 5 个字母为 a、b、c、d、e，设计的哈夫曼编码如下：（答案不唯一）



d: 000
a: 001
e: 01
c: 10
b: 11

（2）二进制表示的等长编码: 000, 001, 010, 011, 100;

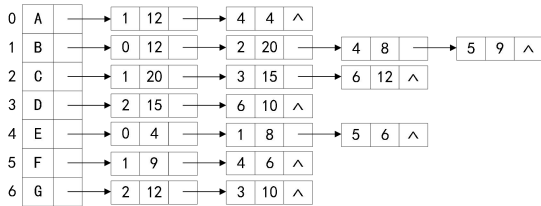
（3）二进制表示的等长编码的长度为 3;

哈夫曼编码的平均长度为: $(2+3)/25 \cdot 3 + (5+7+8)/25 \cdot 2 = 2.2$;

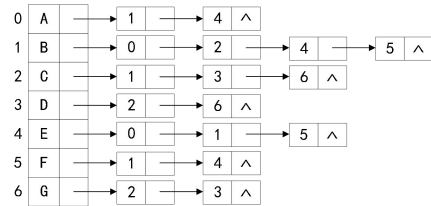
哈夫曼编码的平均长度小，可以有效减少传输电文的总长；而设计 0~4 的二进制表示的等长编码相对简单。

3、（7 分）

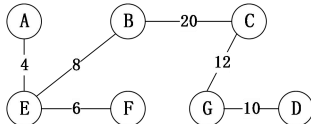
（1）该图对应的邻接表存储结构示意图如下:



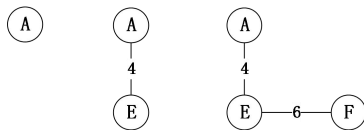
或



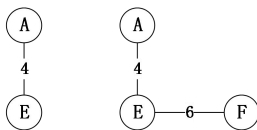
（2）修建公路的费用最节约的方案如下:



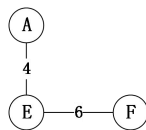
参考过程如下:



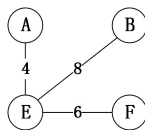
(0)



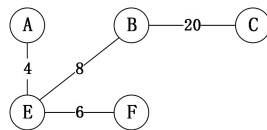
(1)



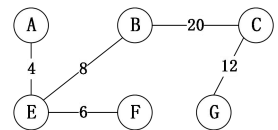
(2)



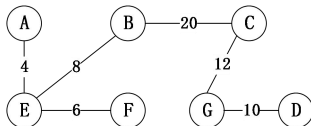
(3)



(4)



(5)



(6)

4、（7分）

（1）事件的最早、最晚发生时间：

事件 V_i	最早发生时间 $Ve(i)$	最晚发生时间 $VI(i)$
V_0	0	0
V_2	15	15
V_1	19	19
V_3	29	29
V_4	28	30
V_5	35	35

事件 a_i	最早开始时间 $e(i)$	最晚开始时间 $l(i)$
a_1	0	17
a_2	0	0
a_3	15	15
a_4	19	19
a_5	19	21
a_6	15	19
a_7	29	29
a_8	28	30

（2）活动的最早、最晚开始时间：

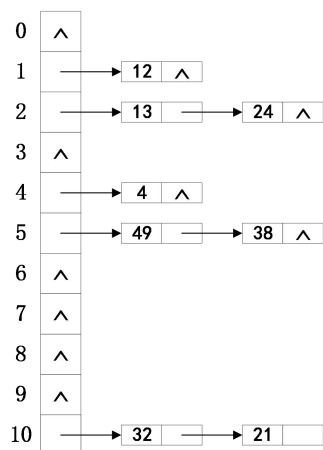
（3）完成此工程，至少需要的时间为 35；关键活动有： a_2 、 a_3 、 a_4 、 a_7 ；关键路径为： $V_0V_2V_1V_3V_5$ 。

5、（5分）

（1）哈希查找：

关键字	32	13	49	24	38	21	4	12
地址	10	2	5	2	5	10	4	1

构造的哈希表如下：

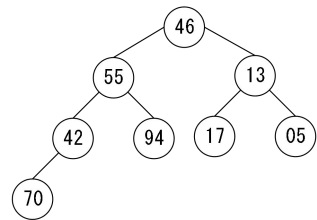


（2） $ASL_{succ} = (1*5+2*3) / 8=11/8$

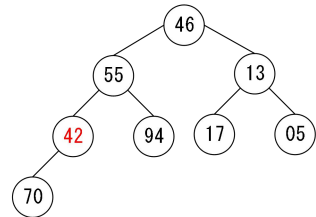
（3） $ASL_{unsucc} = (1+2+3+1+2+3+1+1+1+3) / 11=19/11$

6、（5分）待排序：（46，55，13，42，94，17，05，70），建立初始堆（小顶堆）的过程：

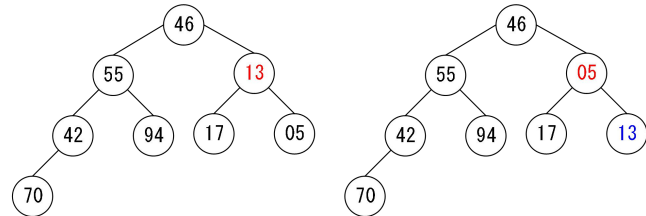
(0) 无序序列



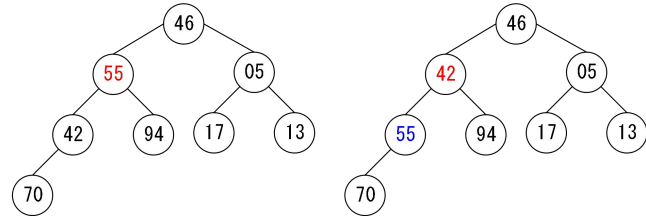
(1) 42向下筛选



(2) 13向下筛选



(3) 55向下筛选



(4) 46向下筛选

