

一、填空题（每空1分，共10分）

1. 向一个长度为 n 的向量的第 i 个元素 ($1 \leq i \leq n+1$) 之前插入一个元素时，需向后移动 $n-i+1$ 个元素。
2. 带有一个头结点的单链表head为空的条件是 `head->next=head`。
3. 由带权为9、2、5、7的四个叶子结点构造一棵哈夫曼树，该树的带权路径长度为 44。
4. 设 $s='I AM A TEACHER'$ ，其长度是 14。
5. 二维数组A[10][20]采用列序为主方式存储，每个元素占一个存储单元并且A[0][0]的存储地址是200，则A[6][12]的地址是 326。
6. 假设在有序线性表A[1..20]上进行二分查找，则比较一次查找成功的结点数目个数为 1。
7. 深度为k的完全二叉树至少有 2^{k-1} 个结点，至多有 2^k-1 个结点。
8. 在无向图G的邻接矩阵A中，若 $A[i][j]$ 等于1，则 $A[j][i]$ 等于 1。
9. 执行广义表操作 `GetTail[GetHead[((a,b),(c,d))]]` 后的结果为 `(b)`。

二、选择题（每小题1分，共10分）

1. 下列数据中，哪一个是非线性结构？(**D**)
A. 栈 B. 队列 C. 字符串 D. 二叉树
2. 设P指向单链表的某个节点，在P之后插入一个节点S的操作为(**C**)
A. $P->next = S;$
B. $P->next = S->next; S->next = P->next;$
C. $S->next = P->next; P->next = S;$
D. $P->next = S; P = P->next;$
3. 设三个元素的进栈顺序为abc，则下列哪个序列是不可能出现的出栈序列
(**D**)
A. abc B. acb C. cba D. cab
4. 假设用块链存储结构表示串，如果每一个块的大小为4个字符（每个字符占用一个字节），一个指针占4个字节，则一个长为15的串需要多少字节的存储空间 (**C**)
A. 15 B. 20 C. 32 D. 40
5. 用一维数组存储一个 6×6 的对称阵A，以行序为主序存储该矩阵的下三角（包括对角线）元素，数组的起始位置为100，每个数组元素占4个字节，那么A[3]

][6]的存储位置为(A)

- A. 168 B. 140 C. 132 D. 108

6. 设一个度为3的树中，度为1, 2, 3的结点个数分别为2, 3, 4，则该树中有多少个叶子结点？(D)

- A. 9 B. 10 C. 11 D. 12

7. 一个AOV网如图1所示，则下列哪一个序列不是该图的拓扑有序序列？(B)

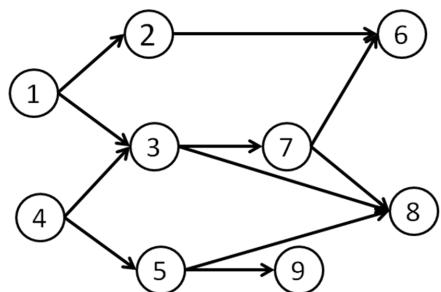


图 1

- A. 124359786 B. 123459786
C. 142357896 D. 413257869

8. 对一个长度为10的有序表进行折半查找，在等概率情况下查找成功的平均查找长度为(A)

- A. 2.9 B. 3.0 C. 3.2 D. 3.4

9. 下列排序算法中，那个算法是稳定的？(D)

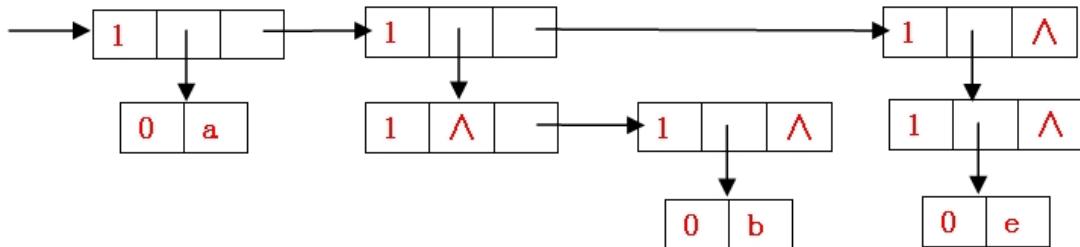
- A. 希尔排序 B. 快速排序 C. 堆排序 D. 归并排序

10. 设数据结构A = (D, R)。其中D={a, b, c, d, e}，R={⟨a, b⟩, ⟨a, c⟩, ⟨b, d⟩, ⟨b, e⟩}。则数据结构A是(B)

- A. 线性结构 B. 树型结构 C. 集合 D. 网状结构

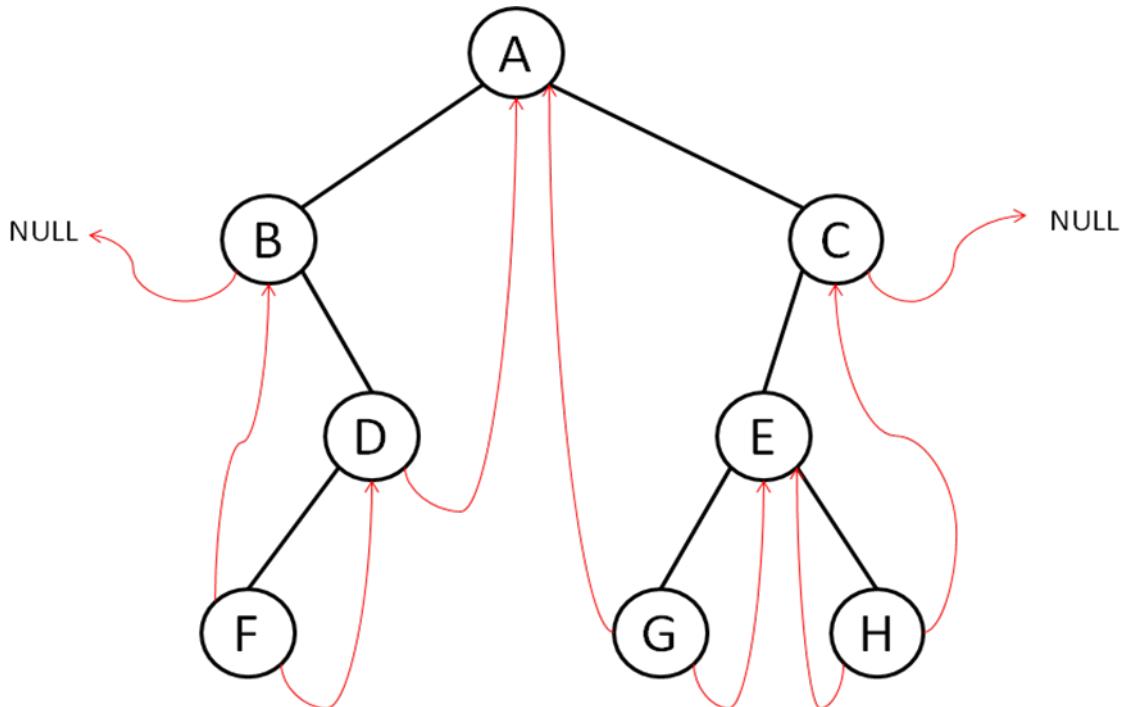
三. 简答题（每小题分值见题首，共40分）

1. (6分) 设广义表 L= (a, ((), b), (e))，画出L的头尾链表存储结构。



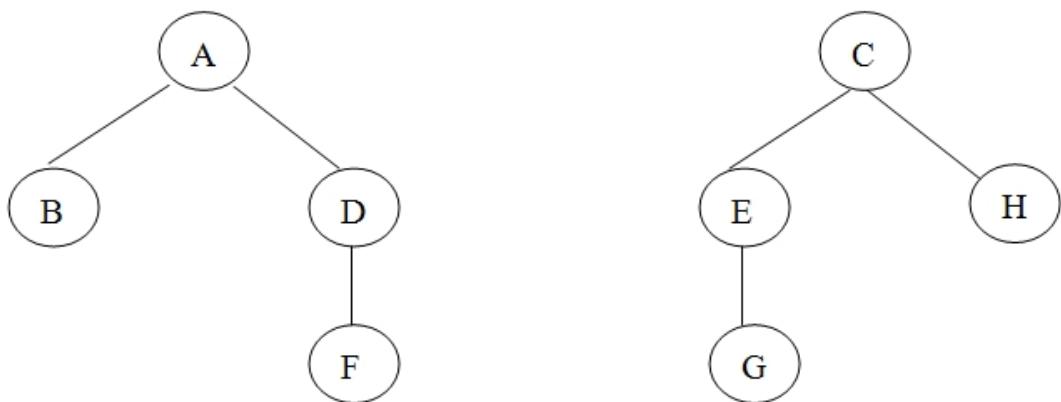
2. (10分) 设某二叉树的先序序列ABDFCEGH，中序序列BFDAGEHC

(1) 画出对应的二叉树; 4分



(2) 在(1)的二叉树中补出中序线索(请将线索画成带箭头的虚线); 3分

(3) 将这棵二叉树转换成树或森林, 并画出来。3分



3. (14分) 设加权有向图的如图2所示:

45

3

25

21

20

15

20

35

20

30

图 2

(1) 写出该图的邻接矩阵 (顶点按A, B, C, D, E的顺序) ; 3分

$$A = \begin{bmatrix} 0 & 3 & 2 & 0 & \infty & 4 & 5 \\ \infty & 0 & 1 & 5 & 2 & 5 \\ 2 & 1 & 0 & 0 & 0 & 2 & 0 \\ \infty & 2 & 0 & 0 & 3 & 5 \\ 0 & \infty & \infty & 3 & 0 & 0 \end{bmatrix}$$

(2) 写出以A为起点的深度优先遍历序列; 3分

DFS序列: A B C D E

(3) 写出以A为起点的广度优先遍历序列; 3分

BFS序列: A B C E D

(4) 利用Dijkstra算法求出A到其余各点的最短路径 (填表1, 并写出最短路径长度)。5分

表1

顶点	已确定的最短路径顶点集合	未确定的最短路径顶点集合	Distance			
			3	20	∞	45
初态	{A}	{B, C, D, E}	3	20	∞	45
B	{A, B}	{C, D, E}	3	18	∞	28
C	{A, B, C}	{D, E}		18	38	28
E	{A, B, C, E}	{D}			38	28
D	{A, B, C, E, D}	Φ			38	

由此可知: A到B的最短距离是3; A到C的最短距离是18; A到D的最短距离是38; A到E的最短距离是28。

4. (10分) 设关键字序列为 {26, 36, 41, 38, 44, 15, 68, 12, 06, 51, 25} , 哈希表长度为15, 即A[0..14], 哈希函数为 $H(key) = key \% 13$ (1) 求装填因子 α ; 1分

$$\alpha = 11/15 = 0.73$$

(2) 用线性探测再散列解决冲突, 构造哈希表 (填表2); 7分

(3) 求等概率下查找成功的平均查找长度ASL。2分

表2

哈希地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
比较次数	1	5	1	2	2	1	1				1		1	2	3
关键字值	26	25	41	15	68	44	06				36		38	12	51
冲突次数	0	4	0	1	1	0	0				0		0	1	2

ASLsuccess
1 6 2 3 3 1 8 1 1 8 2

四. 算法分析题 (每小题分值见题首, 共20分)

1. (6分) 试分析链表L的结构并简述以下算法的功能。

```
LinkedList AR(LinkedList L, ElemtType x)
{
    LNode *pre=null,*q;
    if(!L)    return L;
    q=L;
    while(q->next !=L && q->data !=x)
    {
        pre=q;      q=q->next;
    }
    if(q->data!=x)      return L;
    else
    {
        if(q!=L){
            pre->next=q->next;free(q);
        }
        else
            if(L->next==L){
                free(q);L=null;
            }
        else{
            pre=L->next;
            while(pre->next !=q)
                pre=pre->next;
            L=L->next;
            pre->next=L;
            free(q);
        }
    }
    return L;
}
```

答案:

1. L为不带头结点的单向循环链表。(3分)
2. 该算法的功能是删除L中数据元素值为x的结点。(3分)

2. (6分) 试分析链表L的结构并简述以下算法的功能。

```
LinkedList AE (LinkedList &L)
```

```
{
    LNode *p,*r;
```

```

p=L->next;
L->next=null;
while (p!=null) {
    r=p->next;
    p->next=L->next;
    L->next=p;
    p=r;
}
return L;
}

```

答案：

1. L为带头结点的单向链表。 (3分)
2. 该算法的功能是将L中的数据元素就地逆置。 (3分)

3. (4分) 试简述下面算法的功能。

```

void AI (int data )
{
    STACK S;
    InitStack (&S);
    while (data) {
        Push (&S, data%2);
        data/=2;
    }
    while (!StackEmpty (S)) {
        Pop (&S, &data);
        printf( "%d" , data);
    }
}

```

参考答案： 该算法的功能是借助栈S将十进制数data转换为二进制数。

4. (4分) 试简述下面算法的功能。

```

void AL(Queue &Q)
{
    Stack S; int d;
    InitStack (S);
    while (!QueueEmpty (Q)) {
        DeQueue (Q, d);
        Push (S, d);
    }
    while (!StackEmpty (S)) {
        Pop (S, d);
        EnQueue (Q, d);
    }
}

```

}

参考答案：该算法的功能是借助栈S将队列Q的头尾颠倒。

五. 算法设计题（每小题10分，共20分）

1. 下面的算法功能是在二叉排序树 T 中插入一个数据元素x，请在空缺处填上合适的语句，使算法完整。

```
typedef struct node{  
    ElemType data;  
    struct node *lchild,*rchild;  
}Node, *BiTree;  
  
void Insert(BiTree &T, ElemType x){  
    p=(BiTree)malloc(sizeof(Node));  
    j_____;  
    p->lchild=p->rchild=NULL;  
    if(!T) k_____;  
    else  
    {  
        q=T;  
        while(q->lchild!=NULL || q->rchild!=NULL)  
        {  
            if(x<q->data)  
            { if(q->lchild!=NULL) l_____;  
              else break;  
            }  
            else  
            {  
                if(q->rchild!=NULL) q=q->rchild;  
                else break;  
            }  
        }  
        if(x<q->data) m_____;  
        else n_____;  
    }  
}
```

j p->data=x

k T=p

l q=q->lchild

mq->lchild=p

nq->rchild=p

说明：每个空格占2分。

2. 写出快速排序的一趟划分算法。

int Partition(SqList &L,int low,int high){ //一趟快速排序

//交换子表 r[low...high]的记录，使支点（枢轴）记录到位，并返回其位置

。返回时，//在支点之前的记录均不大于它，支点之后的记录均不小于它。

L.r[0]=L.r[low]; pivotkey=L.r[low].key; 1分

while(low < high){ 1分

while(low<high && L.r[high].key>=pivotkey)--high; 2分

L.r[low]=L.r[high]; 1分

while(low<high && r[low].key<=pivotkey) ++low; 2分

L.r[high]=L.r[low]; 1分

}

L.r[low]=L.r[0]; 1分

return low; 1分

} //Partition