武汉科技大学

2016 年攻读硕士学位研究生入学考试试题

科目名称：数据结构（C 语言版）（■A 卷□B 卷）科目代码：856

考试时间：3 小时　　满分 150　分

可使用的常用工具：√无　□计算器　□直尺　□圆规（请在使用工具前打√）

注意：所有答题内容必须写在答题纸上，写在试题或草稿纸上的一律无效；考完后试题随答题纸交回。

一、选择题(共 10 小题，每小题 2 分，共 20 分)

1. 以下说法正确的是（　　）。
   A）数据元素是数据的最小单位
   B）数据项是数据的基本单位
   C）数据结构是带有结构的各数据项的集合
   D）一些表面上很不相同的数据可以有相同的逻辑结构
2. 在顺序表（长度为 127）中插入一个元素平均要移动（　　）个元素。
   A）8　　B）63.5　　C）63　　D）7
3. 若完全二叉树的结点总数为 1001，则度为 1 的结点有（　　）个。
   A）0　　B）1　　C）500　　D）501
4. 二叉树先序遍历 x 在 y 之前，后序遍历 x 在 y 之后，则 x 是 y 的（　　）。
   A）左兄弟　　B）右兄弟　　C）祖先　　D）后裔
5. 二叉树在线索化后，仍不能有效求解的问题是（　　）。
   A）前序线索二叉树中求前序后继　B）中序线索二叉树中求中序后继
   C）中序线索二叉树中求中序前驱　D）后序线索二叉树中求后序后继
6. 下列关于 AOE 网的叙述中，不正确的是（　　）。
   A）某些关键活动提前，则整个工程将会提前完成
   B）任一关键活动提前，则整个工程将会提前完成
   C）所有关键活动提前，则整个工程将会提前完成
   D）关键活动不按期完成会影响整个工程的完成时间
7. 12 个数据有序顺序存储，采用二分查找，查找失败时的 ASL 值是（　　）。
   A）37/12　　B）63/13　　C）39/12　　D）49/13
8. 二叉查找树的查找效率与二叉树的（　　）有关。
   A）高度　　B）结点的多少　　C）树型　　D）结点的位置
9. 用函数 H(k)=key%17 构造散列表，则链地址法解决冲突需（　　）个链表。
   A）17　　B）13　　C）16　　D）任意
10. 在快速排序过程中，下列结论正确的是（　　）。

A）左、右两个子表都已各自排好序　B）左边的元素都不大于右边的元素
C）左边子表长度小于右边子表长度　D）左、右两边元素的平均值相等

二、填空题(共 10 小题，每小题 2 分，共 20 分)

1. 数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的（　　）等的学科。
2. 在单链表（长度为 n）给定值 x 的结点后插入新结点的时间复杂度为（　　）。
3. 判断表达式中左右括号是否配对的算法采用（　　）数据结构最佳
4. 设广义表 L=((a,b,c))，则 L 的长度为（　　）。
5. 由 4 个结点可以构造出（　　）种不同的二叉树。
6. 用数组 A[0…n-1]存储完全二叉树，则 A[i]的右子女是结点（　　）。
7. 在一个图中，所有顶点的度数之和等于所有边数的（　　）倍。
8. 为了实现图的广度优先搜索，除了一个标志数组标志已访问的结点外，还需（　　）存放被访问的结点以实现遍历。
9. 求图中一个顶点到其它各个顶点最短路径的算法是（　　）算法。
10. 具有 12 个记录的序列，采用冒泡排序最少的比较次数是（　　）。

三、综合应用题(共 7 小题，每小题 10 分，共 70 分)

1. 将三对角矩阵 A[1..n，1..n]的非零元素逐行存放于数组 B[0..3n-3]中，使得 B[k] =A[i，j]，求：
（1）用 i，j 表示 k 的变换公式　　（2）用 k 表示 i，j 的变换公式
2. 设二叉树的顺序存储结构如下：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
e	a	f		d		g			c	j			h	i					b

（1）画出该二叉树的逻辑结构
（2）写出其先序、中序、后序序列
（3）画出其后序线索二叉树
（4）把它转换成对应的森林
3. 给定序列(26,25,20,33,21,24,45,204,42,38,29,31)，要用散列法进行存储，散列函数采用除留余数法，用线性探测法解决冲突，负载因子为 0.6。
（1）设计哈希函数
（2）画出哈希表
（3）计算等概率情况下查找成功和失败的平均查找长度
4. 对有序表(31,34,45,57,64,70,72,84,88,91,97,105,124)折半查找，要求
（1）画出描述折半查找过程的判定树；
（2）若查找元素 91，需依次与那些元素比较?
（3）若查找元素 30，需依次与那些元素比较?
（4）分别求等概率情况下查找成功和不成功时的平均查找长度。
5. 已知关键字序列（40，35，61，87，72，16，25，50），
（1）写出用快速排序方法升序排列该序列一趟后的结果
（2）写出用堆排序进行升序排列时的初始堆

（3）写出堆排序 1 趟以后（交换与调整之后）的结果

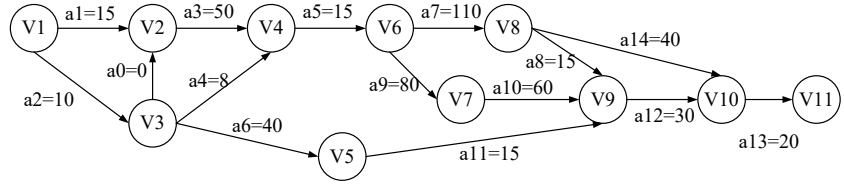
（4）写出 1 趟冒泡排序后的结果

（5）写出 1 趟归并排序后的结果

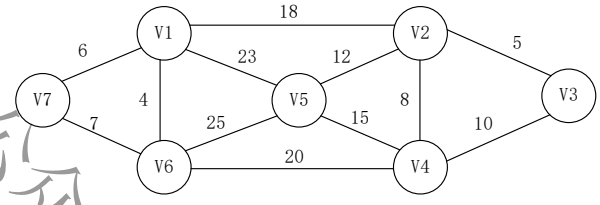
6. 有以下 AOE 网：

（1）求各事件的最早/迟发生时间　　（2）求各活动的最早/迟开始时间

（3）给出其关键路径　　（4）其拓扑序列共有多少种



7. 分别用 Prim 和 kruskal 算法构造最小生成树。（需标示每一步构造过程）



四、算法设计(第 1 题 10 分，第 2，3 题各 15 分，共 40 分)

1. 设计算法，在不带头结点的单链表 L 上实现删除 data 域值为 x 的所有结点，返回删除结点的个数；

```
int  List_Delete（struct  LNode  *L，KeyType  x）
```

2. 采用链式存储实现栈的操作（数据元素类型为 ElemType），包括栈的初始化 InitStack、入栈 Push、出栈 Pop、取栈顶元素 Peek、判栈空 Empty、清空栈 ClearStack 以及返回栈中元素个数 GetLen 等操作，并作简单注释。

3. 根据给定的 n 个权值可以构造一颗哈夫曼树。若哈夫曼树采用顺序存储结构，每个结点的数据结构采用如下格式。

```
Typedef  struct
{ unsigned int weight;  //结点的权值
  unsigned int parent,lchild,rchild;//分别存放双亲、左右孩子的下标
} Huffman;
```

试设计如下算法 CreatHuffman，根据给定的 n 个权值构造一颗哈夫曼树。

```
void  CreatHuffman(Huffman HT[], int n, int w[]);
```

其中：HT 为构造的哈夫曼树，n 表示权值个数，w 用来存储所有权值

下图是根据权值 7，5，2，4 所构造出来的哈夫曼树（-1 表示空）。

	lchild	weight	rchild	paraent
0	-1	7	-1	6
1	-1	5	-1	5
2	-1	2	-1	4
3	-1	4	-1	4
4	2	6	3	5
5	1	11	4	6
6	0	18	5	-1

参考答案(A)

一、选择题（10 小题，每题 2 分，共 20 分）

DBACD ADCAB

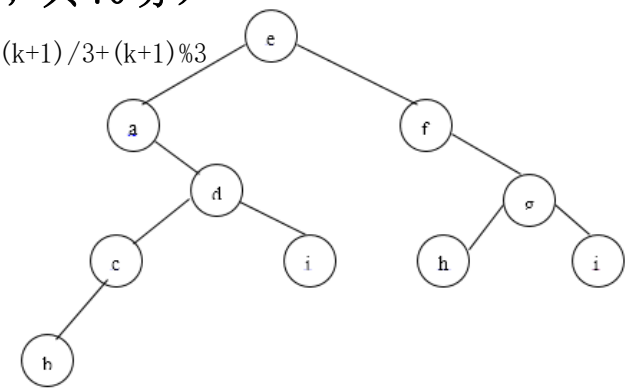
二、填空题（10 小题，每题 2 分，共 20 分）

1. 关系和操作　2. O(n)　3. 栈　4. 1　5. 14
6. A[2i+2]　7. 2　8. 队列　9. Dijkstra　10. 11

三、综合应用题（7 小题，每题 10 分，共 70 分）

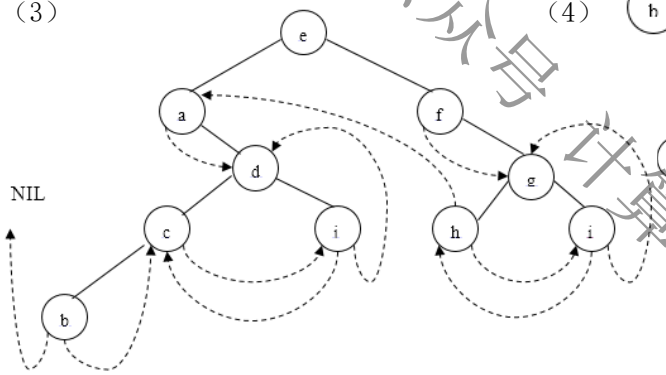
1. （1）k=2i+j-3　（2） i=(k+1)/3+1　j=(k+1)/3+(k+1)%3

2. （1）

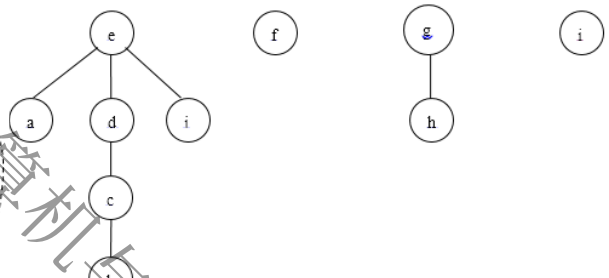


（2）先序：eadcbjfghi
中序：abcdjefhgi
后序：bcjdahigfe

（3）



（4）



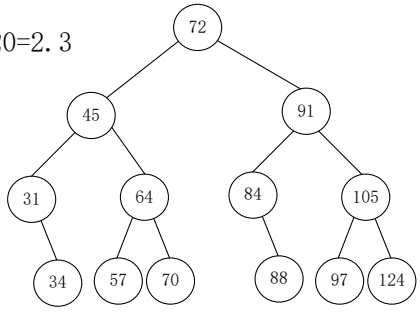
3. 散列函数 H(k)=k%19

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
38	20	21		42	24	25	26	45		29		31		33	204				
1	1	1		1	1	1	1	2		1		1		1	2				

成功：ASL=14/12=7/6

不成功：ASL=(4+3+2+1+6+5+4+3+2+1+2+1+2+1+3+2+1+1+1+1)/20=46/20=2.3

4. （1）　（2） 72　91　（3） 72　45　31



（4）查找成功的平均查找长度：（1+2*2+4*3+6*4）/13=41/13
不成功时的平均查找长度：(2*3+12*4)/14=54/14=27/7

5. （1）快速排序一趟后的结果：25　35　16　40　72　87　61　50
（2）堆排序进行升序初始堆：87　72　61　50　40　16　25　35
（3）堆排序 1 趟以后的结果：72　50　61　35　40　16　25　87
（4）1 趟冒泡排序后的结果：35　40　61　72　16　25　50　87
（5）1 趟归并排序后的结果：35　40　61　87　16　72　25　50

6. （1）

事件	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
最早发生时间	0	15	10	65	50	80	160	190	220	250	270
最迟发生时间	0	15	15	65	205	80	160	205	220	250	270

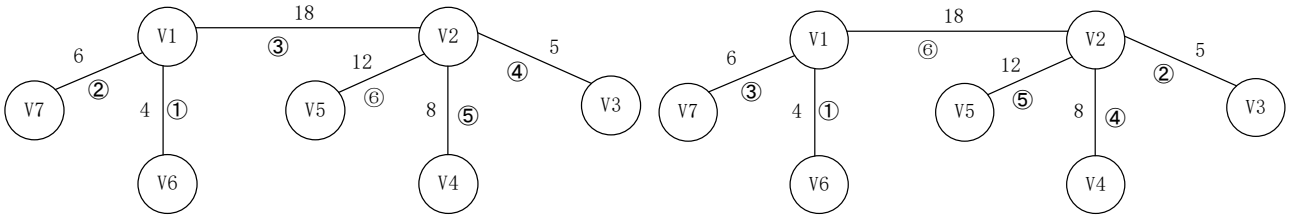
（2）

活动	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14
e	10	0	0	15	10	65	10	80	190	80	160	50	220	250	190
l	15	0	5	15	57	65	165	95	205	80	160	205	220	250	210

（3）关键路径：a1->a3->a5->a9->a10->a12->a13

（4）12

7.



四、算法设计（第1题10分，第2，3题各15分，共40分）

1.

```
int List_Delete(struct LNode *L,KeyType x)
{ int c=0;  if(L==NULL)  return 0;  p=L;
  while(p->next)
  { if(p->next->data==x){ c++;s=p->next; p->next=s->next; free(s); } else  p=p->next;  }
    if(L->data==x) {   c++;    p=L;    L=L->next;    free(p);   }
    return c;
}
```

2.

```
typedef struct node{   ElenType data;    struct node *next;} LinkStack;
void InitStack(LinkStack &L)     {    L=NULL;   }
void Push(LinkStack &L,ElemType x)
{   p=(LinkList *)malloc(sizeof(LinkList));   p->data=x;   p->next=L;    L=p;  }
void Pop(LinkStack &L,ElemType &x){ if(L==NULL) return;  p=L;  L=L->next; x=p->data; free(p);}
void Peek(LinkStack &L,ElemType &x){  if(L==NULL) return;    x=L->data;   }
int Empty(LinkStack L){    return L==NULL;    }
void ClearStack(LinkStack &L){    while(L)  {   p=L->next;    free(L);    L=p;  }      }
int GetLen(LinkStack L){  int i=0;    p=L;   while(p) {   i++;    p=p->next;  }    return i;}
```

3.

```
void  CreatHuffman (Huffman  HT [ ], int n , int  w [ ] )
{  for (i=0; i<n; i++)
   {  HT[i].lchild=-1;   HT[i].rchild=-1;  HT[i].weight=w[i];  HT[i].parent=-1; }
   for (i=n; i<2*n-1; i++)
   { search(HT,i-1,s1,s2);  HT[i].weight=HT[s1].weight+HT[s2].weight;
     HT[i].lchild=s1;      HT[i].rchild=s2;     HT[i].parent=-1;       }
 }
```