

北京交通大学 2015 年硕士研究生入学考试试卷

科目代码: 925 科目名称: 数据结构

共 7 页, 第 1 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!

一、填空题 (本大题共 15 小题, 每小题 2 分, 共 30 分)

1. 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 $T='abcaababcbcab'$, 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_____个关键字。
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 $O(\quad)$ 。

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. 已知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

北京交通大学 2015 年硕士研究生入学考试试卷

科目代码: 925 科目名称: 数据结构

共 7 页, 第 2 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!

针 s 指向的新结点, 相应操作为 ()。

- A. $p \rightarrow next = s; s \rightarrow last = p; p \rightarrow next \rightarrow last = s; s \rightarrow next = p \rightarrow next$
- B. $p \rightarrow next = s; p \rightarrow next \rightarrow last = s; s \rightarrow last = p; s \rightarrow next = p \rightarrow next$
- C. $s \rightarrow last = p; s \rightarrow next = p \rightarrow next; p \rightarrow next = s; p \rightarrow next \rightarrow last = s$
- D. $s \rightarrow last = p; s \rightarrow next = p \rightarrow next; p \rightarrow next \rightarrow last = s; p \rightarrow next = s$

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, $e_1, e_2, e_3, e_4, e_5, e_6$ 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. $e_2, e_4, e_3, e_5, e_1, e_6$
- B. $e_2, e_5, e_1, e_3, e_4, e_6$
- C. $e_5, e_1, e_6, e_3, e_2, e_4$
- D. $e_4, e_1, e_3, e_5, e_2, e_6$

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
- C. 左子树中最左下的结点
- D. 左子树中最右下的结点

10. 对某个无向图的邻接矩阵来说, 下列哪种说法是正确的 ()。

北京交通大学 2015 年硕士研究生入学考试试卷

科目代码: 925 科目名称: 数据结构

共 7 页, 第 3 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!

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 V_i 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 $\text{Head}(\text{Head}(\text{Tail}(A)))$ 结果为 ()。

A. x

B. (a, b)

C. $(x, (a, b))$

D. A

13. 若一个栈以向量 $V[1..n]$ 存储, 栈内有 s 个元素 ($s < n$), 栈顶指针 top 为 $s+1$, 则下面 x 进栈的正确操作是()。

A. $\text{top} := \text{top} + 1; V[\text{top}] := x$

B. $V[\text{top}] := x; \text{top} := \text{top} + 1$

C. $\text{top} := \text{top} - 1; V[\text{top}] := x$

D. $V[\text{top}] := x; \text{top} := \text{top} - 1$

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 S_1 和 S_2 , 求 S_2 在 S_1 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 V_i 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。()

8. 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。()

9. 哈希表的平均查找长度与处理冲突的方法无关。()

北京交通大学 2015 年硕士研究生入学考试试卷

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 7 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。

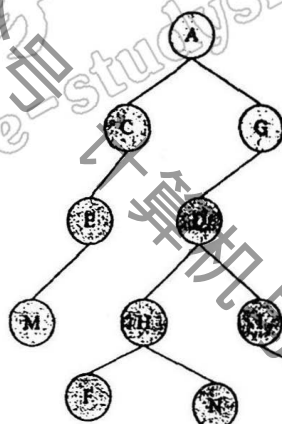


图 4-1 孩子兄弟链表表示的二叉树

2. 表长度为 13, 数组的存储地址为 $a[0]$ 至 $a[12]$, 哈希函数为 $H(\text{key}) = \text{key} \bmod 13$, 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 $n_3 = (n_0 - 1) / 2$, 其中 n_3 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 V_1 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。

北京交通大学 2015 年硕士研究生入学考试试卷

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!

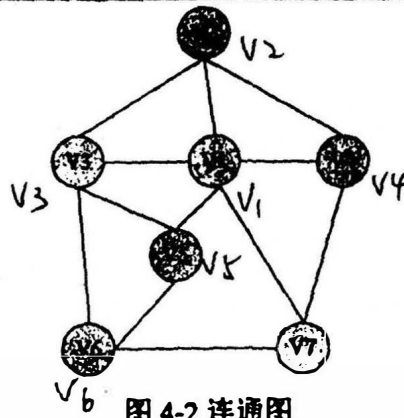


图 4-2 连通图

7. 在图 4-3 的带权有向图中, 求从顶点 V_0 到其余各个顶点的最短路径, 给出过程。

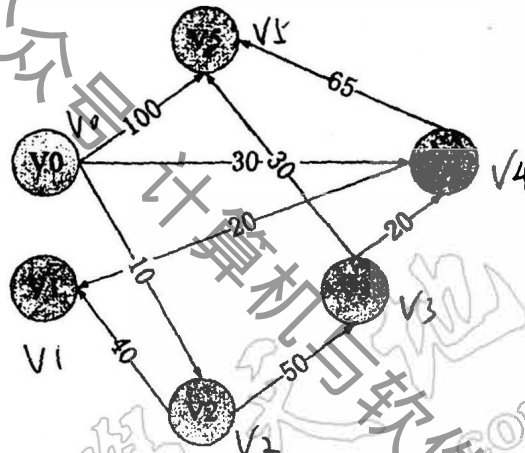


图 4-3 带权有向图

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 (LinkedList &L) {
    t=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) {
        p->next= (1); //让 p 结点 next 域指针指向其前驱
        t=p;         //顺链向后移动指针 t
        p=q;         //顺链向后移动指针 p
        q= (2);      //顺链向后移动指针 q
    }
}
```

北京交通大学 2015 年硕士研究生入学考试试卷

科目代码: 925 科目名称: 数据结构

共 7 页, 第 6 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!

```
L->next= (3); //让 L 的 next 域指针指向新链表的第一个结点
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
    if (!T)    depthval = (4);
    else {
        depthLeft = Depth( T->lchild );
        depthRight = Depth( T->rchild );
        depthval = (5) + (depthLeft > depthRight ?
            (6));
    }
    return depthval;
}
```

3. 下列算法用于进行一趟希尔排序。

```
void ShellInsert ( SqList &L, int dk ) { //一趟增量为 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);
                j-=dk)
                L.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'};
static int a[n][n]={0,1,1,0,0, 1,0,1,1,0, 1,1,0,1,1, 0,1,1,0,1, 0,0,1,1,0};
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;
for(i=0; i<n; i++) { G[i].vertex=ch[i]; G[i].firstedge=0; }
for(i=0; i<n; i++)
{ for(j=0; j<n; j++)
```


北京交通大学 2015 年硕士研究生入学考试试卷

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][j]==1)
      { s=(EdgeNode*)malloc(sizeof(EdgeNode));
        s->adjvex=j; s->next=G[i].firstedge; G[i].firstedge=s;
      }
    }
}

void MS(int i)
{ EdgeNode *p;
  printf(" %c", G[i].vertex); visited[i]=1; p=G[i].firstedge;
  while(p)
  { if(visited[p->adjvex]==0) MS(p->adjvex);
    p=p->next;
  }
}

void MSTO
{ int k;
  for(k=0;k<n;k++) visited[k]=0;
  for(k=0;k<n;k++)
    if(visited[k]==0) MS(k);
}

main()
{ create(); MSTO; }

```

计算机/软件工程专业

每个学校的

考研真题/复试资料/考研经验

考研资讯/报录比/分数线

免费分享



微信 扫一扫

关注微信公众号

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:

$$M = \begin{bmatrix} 0 & 14 & 0 & 0 & -5 \\ 0 & -7 & 0 & 0 & 0 \\ 36 & 0 & 0 & 28 & 0 \end{bmatrix} \rightarrow T = \begin{bmatrix} 0 & 0 & 36 \\ 14 & -7 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 28 \\ -5 & 0 & 0 \end{bmatrix}$$

#define MAXSIZE 1000 //非零元个数的最大值

typedef struct {

int i, j; //该非零元的行下标和列下标

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

int mu, nu, tu; //矩阵的行数、列数和非零元个数

} 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。