华中师范大学 2005–2006 学年第一学期

期末考试试卷（A 卷）

课程名称 数据结构 课程编号 43010700 任课教师_____

题型	选择题	填空题	判断题	简答题	分析题	应用题	总分
分值	30	15	10	20	10	15	100
得分							

得分	评阅人

一、单项选择题：（共 15 题，每题 2 分）

（ ）1. 算法分析的两个主要方面是_____

A．空间复杂性和时间复杂性　　B．正确性和简明性

C．可读性和文档性　　D．数据复杂性和程序复杂性

（ ）2. 非线性结构是数据元素之间存在一种_____

A．一对多关系　　B．多对多关系

C．多对一关系　　D．一对一关系

（ ）3. 数据在计算机存储器内表示时，物理地址与逻辑地址相同并且是连续的，称之为_____

A．存储结构　　B．逻辑结构

C．顺序存储结构　　D．链式存储结构

（ ）4. 链表是一种采用_____存储结构存储的线性表；

A．顺序　　B．链式　　C．星式　　D．网状

（ ）5. 线性表若采用链式存储结构时，要求内存中可用存储单元的地址_____

A．必须是连续的　　B．部分地址必须是连续的

C．一定是不连续的　　D．连续或不连续都可以

（ ）6. 栈中元素的进出原则是_____

A．先进先出　　B．后进先出　　C．栈空则进　　D．栈满则出

（ ）7. 串是一种特殊的线性表，其特殊性体现在_____

A．可以顺序存储　　B．数据元素是一个字符

C．可以链式存储　　D．数据元素可以是多个字符

（ ）8. 设有两个串 p 和 q，求 q 在 p 中首次出现的位置的运算称作_____

A．连接　　B．模式匹配　　C．求子串　　D．求串长

（ ）9. 不含任何结点的空树_____

A．是一棵树；　　B．是一棵二叉树；

C．是一棵树也是一棵二叉树；　　D．既不是树也不是二叉树

（　）10. 二叉树是非线性数据结构，所以_____

A．它不能用顺序存储结构存储；

B．它不能用链式存储结构存储；

C．顺序存储结构和链式存储结构都能存储；

D．顺序存储结构和链式存储结构都不能使用

（　）11. 具有 n(n>0)个结点的完全二叉树的深度为_____

A．$\lceil \log_2(n) \rceil$　　B．$\lfloor \log_2(n) \rfloor$　　C．$\lfloor \log_2(n) \rfloor+1$　　D．$\lceil \log_2(n)+1 \rceil$

（　）12. 把一棵树转换为二叉树后，这棵二叉树的形态是_____

A．唯一的　　B．有多种

C．有多种，但根结点都没有左孩子　　D．有多种，但根结点都没有右孩子

（　）13. 在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和的_____倍。

A．1/2　　B．1　　C．2　　D．4

（　）14. 有 8 个结点的无向图最多有_____条边。

A．14　　B．28　　C．56　　D．112

（　）15. 排序方法中，从未排序序列中依次取出元素与已排序序列（初始时为空）中的元素进行比较，将其放入已排序序列的正确位置上的方法，称为_____

A．希尔排序　　B．冒泡排序　　C．插入排序　　D．选择排序

得分	评阅人

二、填空题：（共 8 题，每空 1 分）

16. 数据结构被形式地定义为（D, R），其中 D 是_____的有限集合，R 是 D 上的_____有限集合。

17. 一个算法的效率可分为_____效率和_____效率。

18. 线性结构中元素之间存在_____关系，树形结构中元素之间存在_____关系，图形结构中元素之间存在_____关系。

19. 向一个长度为 n 的线性表的第 i 个元素(1≤i≤n+1)之前插入一个元素时，需向后移动_____个元素。

20. 栈是一种特殊的线性表，允许插入和删除运算的一端称为_____。不允许插入和删除运算的一端称为_____。

21. 三元素组表中的每个结点对应于稀疏矩阵的一个非零元素，它包含有三个数据项，分别表示该元素的_____、_____和_____。

22. 有向图 G 用邻接表矩阵存储，其第 i 行的所有元素之和等于顶点 i 的_____。

23. 若要求一个稀疏图 G 的最小生成树，最好用_____算法来求解。

得分	评阅人

三、判断正误题：（共 10 题，每题 1 分）

（　）24. 链表的每个结点中都恰好包含一个指针。

（　）25. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。

（　）26. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。

（　）27. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。

（　）28. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结构。

（　）29. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。

（　）30. 栈和链表是两种不同的数据结构。

（　）31. 若二叉树用二叉链表作存贮结构，则在 n 个结点的二叉树链表中只有 n—1 个非空指针域。

（　）32. 二叉树中每个结点有两棵非空子树或有两棵空子树。

（　）33. 二叉树中每个结点的两棵子树是有序的。

得分	评阅人

四、简答题：（共 4 题，每题 5 分）

34. 简述线性结构与非线性结构的不同点。

35. 描述以下三个概念的区别：头指针、头结点、首结点（第一个元素结点）。在单链表中设置头结点的作用是什么？

36.说明线性表、栈与队的异同点。

37. 一棵度为 2 的树与一棵二叉树有何区别？

得分	评阅人

五、分析题：（共 2 题，每题 5 分）

38. 分析下面算法的时间复杂性

```
x=0;
for( i=1;i<n; i++)
   for(j=1;j<n-i; j++)
x++;
```

算法时间复杂度为：

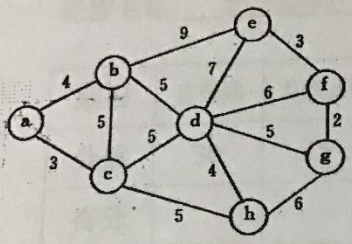
39. 下列三元组表表示一个稀疏矩阵，试写出它所代表的稀疏矩阵。

$$\begin{bmatrix} 6 & 4 & 6 \\ 1 & 2 & 2 \\ 2 & 1 & 12 \\ 3 & 1 & 3 \\ 4 & 4 & 4 \\ 5 & 3 & 6 \\ 6 & 1 & 16 \end{bmatrix}$$

得分	评阅人

六、应用设计题：（共 2 题，共 15 分）

40. 请对下图的无向带权图，写出它的邻接矩阵，并按普里姆算法求其最小生成树。（7 分）



41. 定义二叉树的抽象数据类型，并设计一个求二叉树深度的算法。(8 分)

华中师范大学 2005–2006 学年第一学期
期末考试试卷（A 卷答案）

课程名称 数据结构 课程编号 43010700 任课教师

题型	选择题	填空题	判断题	简答题	分析题	应用题	总分
分值	30	15	10	20	10	15	100
得分							

得分	评阅人

一、单项选择题：（共 15 题，每题 2 分）

1. A 2. B 3. C 4. B 5. D 6. B 7. B 8. B 9. C 10. C
11. C 12. A 13. B 14. B 15. C

得分	评阅人

二、填空题：（共 8 题，每空 1 分）

16. 数据元素 关系
17. 时间 空间
18. 一对一 一对多 多对多
19. n-i+1
20. 栈顶 栈底
21. 行下标 列下标 元素值
22. 出度
23. 克鲁斯卡尔(Kruskal)

得分	评阅人

三、判断正误题：（共 10 题，每题 1 分）

（ × ）24. 链表的每个结点中都恰好包含一个指针。

（ × ）25. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个
前移动。

（ × ）26. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。

（ × ）27. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。

（ √ ）28. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结

（ √ ）29. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。

（ × ）30. 栈和链表是两种不同的数据结构。

（ √ ）31. 若二叉树用二叉链表作存贮结构，则在 n 个结点的二叉树链表中只有 n—1 个非空指

（ × ）32. 二叉树中每个结点有两棵非空子树或有两棵空子树。

（ √ ）33. 二叉树中每个结点的两棵子树是有序的。

得分	评阅人

四、简答题：（共 4 题，每题 5 分）

34. 简述线性结构与非线性结构的不同点。

答：线性结构反映结点间的逻辑关系是 一对一的，非线性结构反映结点间的逻辑关系是多对

35. 描述以下三个概念的区别：头指针、头结点、首结点（第一个元素结点）。在单链表中设置头结
作用是什么？

答：首元结点是指链表中存储线性表中第一个数据元素 a_1 的结点。为了操作方便，通常在链表
结点之前附设一个结点，称为头结点，该结点的数据域中不存储线性表的数据元素，其作用
对链表进行操作时，可以对空表、非空表的情况以及对首元结点进行统一处理。头指针是指
中第一个结点（或为头结点或为首元结点）的指针。若链表中附设头结点，则不管线性表是
表，头指针均不为空。否则表示空表的链表的头指针为空。这三个概念对单链表、双向链表
链表均适用。是否设置头结点，是不同的存储结构表示同一逻辑结构的问题。

简而言之，头指针是指向链表中第一个结点（或为头结点或为首元结点）的指针；
头结点是在链表的首元结点之前附设的一个结点；数据域内只放空表标志和表长等信息（内
针？那还得另配一个头指针!!!）首元素结点是指链表中存储线性表中第一个数据元素 a_1 的

36.说明线性表、栈与队的异同点。

答：相同点：都是线性结构，都是逻辑结构的概念。都可以用顺序存储或链表存储；栈和队列
特殊的线性表，即受限的线性表，只是对插入、删除运算加以限制。

不同点：①运算规则不同，线性表为随机存取，而栈是只允许在一端进行插入、删除运算，因
进先出表 LIFO；队列是只允许在一端进行插入、另一端进行删除运算，因而是先进先出表 F

② 用途不同，堆栈用于子程调用和保护现场，队列用于多道作业处理、指令寄存及其他运算

37. 一棵度为 2 的树与一棵二叉树有何区别？

答：度为 2 的树从形式上看与二叉树很相似，但它的子树是无序的，而二叉树是有序的。即，
树中若某结点只有一个孩子，就无需区分其左右次序，而在二叉树中即使是一个孩子也有左右

得分	评阅人

五、分析题：（共2题，每题5分）

38. 分析下面算法的时间复杂性

```
3.  x=0;
    for(i=1; i<n; i++)
      for (j=1; j<=n-i; j++)
x++;
```

解：因为x++共执行了 n-1+n-2+……＋1=
n(n-1)/2，所以执行时间为 **O（n^2）**

39. 下列各三元组表分别表示一个稀疏矩阵，试写出它们的稀疏矩阵。

(1)
$$\begin{bmatrix} 6 & 4 & 6 \\ 1 & 2 & 2 \\ 2 & 1 & 12 \\ 3 & 1 & 3 \\ 4 & 4 & 4 \\ 5 & 3 & 6 \\ 6 & 1 & 16 \end{bmatrix}$$

解：（1）为6×4矩阵，非零元素有6个。

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ 12 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 6 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix}$$

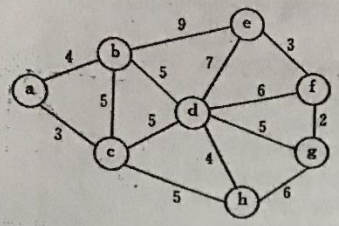
得分	评阅人

六、应用设计题：（共2题，共15分）

40. 请对下图的无向带权图，写出它的邻接矩阵，并按普里姆算法求其最小生成树；（7分）

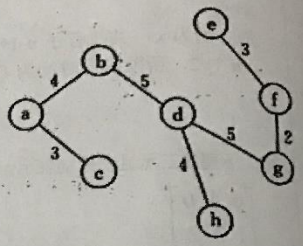
解：设起点为a。可以直接由原始图画出最小生成树，而且最小生成树只有一种（类）！

邻接矩阵为：



$$\begin{bmatrix} 0 & 4 & 3 & \infty & \infty & \infty & \infty & \infty \\ 4 & 0 & 5 & 5 & 9 & \infty & \infty & \infty \\ 3 & 5 & 0 & 5 & \infty & \infty & \infty & 5 \\ \infty & 5 & 5 & 0 & 7 & 6 & 5 & 4 \\ \infty & 9 & \infty & 7 & 0 & 3 & \infty & \infty \\ \infty & \infty & \infty & 6 & 3 & 0 & 2 & \infty \\ \infty & \infty & \infty & 5 & \infty & 2 & 0 & 6 \\ \infty & \infty & 5 & 4 & \infty & \infty & 6 & 0 \end{bmatrix}$$

最小生成树→



41．写出求二叉树深度的算法，先定义二叉树的抽象数据类型。（8 分）

答：设计思路：只查后继链表指针，若左或右孩子的左或右指针非空，则层次数加 1；否则函数返回。但注意，递归时应当从叶子开始向上计数，否则不易确定层数。

算法如下：

```
int depth(liuyu*root)        /*统计层数*/
{    int d,p;                          /*注意每一层的局部变量 d,p 都是各自独立的*/
     p=0;
     if(root==NULL)return(p);          /*找到叶子之后才开始统计*/
      else
      {
         d=depth(root->lchild);
         if(d>p) p=d;                  /*向上回朔时，要挑出左右子树中的相对大的那个深度值*/
        d=depth(root->rchild);
        if(d>p)p=d;
      }
   p=p+1;
   return(p);
}
```

算法二：

```
int Get_Sub_Depth(Bitree T,int x)//求二叉树中以值为 x 的结点为根的子树深度
{
  if(T->data==x)
  {
     printf("%d\n",Get_Depth(T)); //找到了值为 x 的结点,求其深度
     exit 1;
  }
```

```
  else
   {
     if(T->lchild) Get_Sub_Depth(T->lchild,x);
     if(T->rchild) Get_Sub_Depth(T->rchild,x); //在左右子树中继续寻找
   }
 }//Get_Sub_Depth

int Get_Depth(Bitree T)//求子树深度的递归算法
{
  if(!T) return 0;
  else
  {
    m=Get_Depth(T->lchild);
    n=Get_Depth(T->rchild);
    return (m>n?m:n)+1;
  }
}//Get_Depth
```