

华中师范大学 2005-2006 学年第一学期
期末考试试卷 (B 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. 算法分析的目的是_____
- A. 找出数据结构的合理性 B. 研究算法中的输入和输出的关系
C. 分析算法的效率以求改进 D. 分析算法的易懂性和文档性
- () 2. 数据结构中, 与所使用的计算机无关的是数据的_____结构;
- A. 存储 B. 物理 C. 逻辑 D. 物理和存储
- () 3. 链接存储的存储结构所占存储空间_____
- A. 分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针
B. 只有一部分, 存放结点值
C. 只有一部分, 存放表示结点间关系的指针
D. 分两部分, 一部分存放结点值, 另一部分存放结点所占单元数
- () 4. 单链表的存储密度_____
- A. 大于 1; B. 等于 1; C. 小于 1; D. 不能确定
- () 5. 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
- () 6. 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

- () 7. 有 8 个结点的无向连通图最少有 _____ 条边。
A. 5 B. 6 C. 7 D. 8
- () 8. 有 8 个结点的有向完全图有 _____ 条边。
A. 14 B. 28 C. 56 D. 112
- () 9. 用邻接表表示图进行广度优先遍历时, 通常是采用 _____ 来实现算法的。
A. 栈 B. 队列 C. 树 D. 图
- () 10. 用邻接表表示图进行深度优先遍历时, 通常是采用 _____ 来实现算法的。
A. 栈 B. 队列 C. 树 D. 图
- () 11. 深度优先遍历类似于二叉树的
A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
- () 12. 任何一个无向连通图的最小生成树
A. 只有一棵 B. 一棵或多棵 C. 一定有多棵 D. 可能不存在
- () 13. 对 22 个记录的有序表作折半查找, 当查找失败时, 至少需要比较 _____ 次关键字。
A. 3 B. 4 C. 5 D. 6
- () 14. 链表适用于 _____ 查找
A. 顺序 B. 二分法 C. 顺序, 也能二分法 D. 随机
- () 15. 排序方法中, 从未排序序列中挑选元素, 并将其依次插入已排序序列 (初始时空) 的方法, 称为 _____
A. 希尔排序 B. 归并排序 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. 数据结构是一门研究非数值计算的程序设计问题中计算机的 _____ 以及它们之间的 _____ 和运算等的学科。
17. 在线性结构中, 第一个结点 _____ 前驱结点, 其余每个结点有且只有 1 个前驱结点; 最后一个结点 _____ 后续结点, 其余每个结点有且只有 1 个后续结点。
18. 在图形结构中, 每个结点的前驱结点数和后续结点数可以 _____。
19. 向一个长度为 n 的线性表中删除第 i 个元素 ($1 \leq i \leq n$) 时, 需向前移动 _____ 个元素。
20. 在 n 个结点的单链表中要删除已知结点 $*p$, 需找到它的 _____, 其时间复杂度为 _____。
21. 若要求一个稠密图 G 的最小生成树, 最好用 _____ 算法来求解。
22. 子串的定位运算称为串的模式匹配; _____ 称为目标串, _____ 称为模式。
23. 广义表 $\text{GetHead}[\text{GetTail}[\text{GetHead}[(a,b),(c,d)]]]$ 操作的结果为 _____;
24. 如果 n 个顶点的图是一个环, 则它有 _____ 棵生成树。
25. n 个顶点 e 条边的图, 若采用邻接表存储, 则空间复杂度为 _____。
26. 设有一稠密图 G , 则 G 采用 _____ 存储较省空间。

得分	评阅人

三、判断正误题：（共 10 题，每题 1 分）

- () 27. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。
- () 28. 一个栈的输入序列是 12345，则栈的输出序列不可能是 12345。
- () 29. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。
- () 30. 顺序存储方式只能用于存储线性结构。
- () 31. 线性表的逻辑顺序与存储顺序总是一致的。
- () 32. 栈和队列是一种非线性数据结构。
- () 33. 栈和队列的存储方式既可是顺序方式，也可是链接方式。
- () 34. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。
- () 35. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。
- () 36. 二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。

得分	评阅人

四、简答题：（共 4 题，每题 5 分）

37. 数据结构和数据类型两个概念之间有区别吗？

38. 试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？

39. 说明线性表、栈与队的异同点。

40. 设有编号为 1, 2, 3, 4 的四辆列车, 顺序进入一个栈式结构的车站, 具体写出这四辆列车开出车站的所有可能的顺序。

得分	评阅人

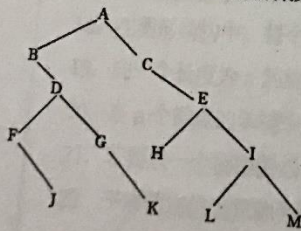
五、分析题: (共 2 题, 每题 5 分)

41. 下列三元组表表示一个稀疏矩阵, 试写出它所表示的稀疏矩阵。

$$\begin{bmatrix} 4 & 5 & 5 \\ 1 & 1 & 1 \\ 2 & 4 & 9 \\ 3 & 2 & 8 \\ 3 & 5 & 6 \\ 4 & 3 & 7 \end{bmatrix}$$

对应的稀疏矩阵为:

42. 试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列。



得分	评阅人

六、应用设计题：(共 2 题，共 15 分)

43. 假设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。试为这 8 个字母设计哈夫曼编码。使用 0~7 的二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。(7 分)

44. 编写按层次顺序(同一层自左至右)遍历二叉树的算法。(8 分)

华中师范大学 2005-2006 学年第一学期
期末考试试卷 (B 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. C 2. C 3. A 4. C 5. B 6. C 7. C 8. C 9. B 10. A
11. A 12. A 13. C 14. A 15. D

得分	评阅人

二、填空题：(共 8 题，每空 1 分)

16. 操作对象 关系
17. 没有 没有
18. 任意多个 。
19. $n-i$
20. 前驱结点的地址 $O(n)$ 。
21. 普里姆(Prim)
22. 被匹配的主串 子串
23. b ；
24. n
25. $O(n+e)$ 。
26. 邻接矩阵

得分	评阅人

三、判断正误题：(共 10 题，每题 1 分)

- (×) 27. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。
- (×) 28. 一个栈的输入序列是 12345，则栈的输出序列不可能是 12345。
- (×) 29. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。
- (×) 30. 顺序存储方式只能用于存储线性结构。
- (×) 31. 线性表的逻辑顺序与存储顺序总是一致的。
- (×) 32. 栈和队列是一种非线性数据结构。
- (√) 33. 栈和队列的存储方式既可是顺序方式，也可是链接方式。
- (√) 34. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底设在这片内存空间的两端。
- (×) 35. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。
- (×) 36. 二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。

得分	评阅人

四、简答题：(共 4 题，每题 5 分)

37. 数据结构和数据类型两个概念之间有区别吗？

答：简单地说，数据结构定义了一组按某些关系结合在一起的数组元素。数据类型不仅定义了一组数据元素，而且还在其上定义了一组操作。

38. 试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？

答：① 顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用连续的地址必须是连续的。

优点：存储密度大（=1），存储空间利用率高。缺点：插入或删除元素时不方便。

② 链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针。

优点：插入或删除元素时很方便，使用灵活。缺点：存储密度小（<1），存储空间利用率低。

顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。

若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；

若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。

39. 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；栈和队列是特殊的线性表，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，因而先出表 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 FIFO。

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

40. 设有编号为 1, 2, 3, 4 的四辆列车, 顺序进入一个栈式结构的车站, 具体写出这四辆列车开出车站的所有可能的顺序。

答: 至少有 14 种。

- ① 全进之后再出情况, 只有 1 种: 4, 3, 2, 1
- ② 进 3 个之后再出的情况, 有 3 种: 3, 4, 2, 1 3, 2, 4, 1 3, 2, 1, 4
- ③ 进 2 个之后再出的情况, 有 5 种: 2, 4, 3, 1 2, 3, 4, 1 2, 1, 3, 4 2, 1, 4, 3 2, 1, 3, 4
- ④ 进 1 个之后再出的情况, 有 5 种: 1, 4, 3, 2 1, 3, 2, 4 1, 3, 4, 2 1, 2, 3, 4 1, 2, 4, 3

得分	评阅人

五、分析题: (共 2 题, 每题 5 分)

41. 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。

$\begin{bmatrix} 4 & 5 & 5 \\ 1 & 1 & 1 \\ 2 & 4 & 9 \\ 3 & 2 & 8 \\ 3 & 5 & 6 \\ 4 & 3 & 7 \end{bmatrix}$

对应的稀疏矩阵为:

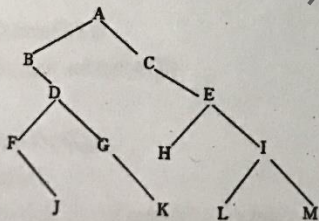
$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 6 \\ 0 & 0 & 7 & 0 & 0 \end{bmatrix}$

42. 试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列。

答: DLR: ABDFJGKCEHILM

LDR: BFJDGKACHELIM

LRD: JFKGDBHLMIECA



得分	评阅人

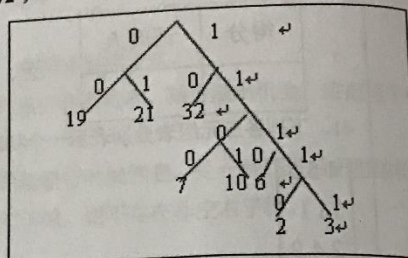
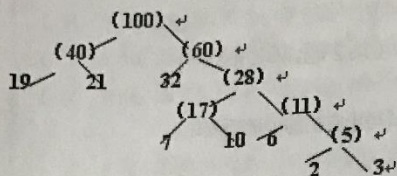
六、应用设计题：(共2题，共15分)

43. 假设用于通信的电文仅由8个字母组成，字母在电文中出现的频率分别为0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。试为这8个字母设计哈夫曼编码。使用0~7的二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。(7分)

解：方案1：哈夫曼编码

先将概率放大100倍，以方便构造哈夫曼树。

$w = \{7, 19, 2, 6, 32, 3, 21, 10\}$ ，按哈夫曼规则：【(2,3), 6], (7,10)], ..., 19, 21, 32

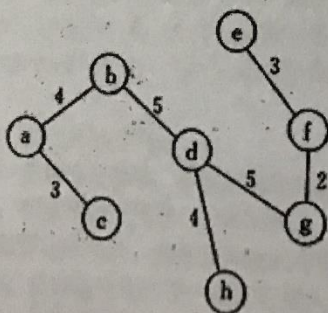


请对下图的无向带权图：

(1) 写出它的邻接表，并按克鲁斯卡尔算法求其最小生成树。

0	4	3	∞	∞	∞	∞	∞
4	0	5	5	9	∞	∞	∞
3	5	0	5	∞	∞	∞	5
∞	5	5	0	7	6	5	4
∞	9	∞	7	0	3	∞	∞
∞	∞	∞	6	3	0	2	∞
∞	∞	∞	5	∞	2	0	6
∞	∞	5	4	∞	∞	6	0

最小生成树→



44. 编写按层次顺序（同一层自左至右）遍历二叉树的算法。（8分）

解：思路：既然要求从上到下，从左到右，则利用队列存放各子树结点的指针是个好办法。

这是一个循环算法，用 while 语句不断循环，直到队空之后自然退出该函数。

技巧之处：当根结点入队后，会自然使得左、右孩子结点入队，而左孩子出队时又会立即使得它的左右孩子结点入队，……以此产生了按层次输出的效果。

level(*T)

/*假设 max 已知*/

```
{
    *T, *p, *q[100];
    int f, r;
    f=0; r=0;          /*置空队*/
    r=(r+1)%max;
    q[r]=T;             /*根结点进队*/
    while(f!=r)         /*队列不空*/
    {
        f=(f+1)%max;
        p=q[f];         /*出队*/
        printf("%d", p->data); /*打印根结点*/
        if(p->lchild)
        {
            r=(r+1)%max; q[r]=p->lchild; /*若左子树不空，则左子树进队*/
        }
        if(p->rchild)
        {
            r=(r+1)%max; q[r]=p->rchild; /*若右子树不空，则右子树进队*/
        }
    }
    return(0);
}
```

法二：

void LayerOrder(Bitree T) //层序遍历二叉树

```
{
    InitQueue(Q); //建立工作队列

    EnQueue(Q, T);
    while(!QueueEmpty(Q))
    {
        DeQueue(Q, p);
        visit(p);
        if(p->lchild) EnQueue(Q, p->lchild);
        if(p->rchild) EnQueue(Q, p->rchild);
    }
} //LayerOrder
```