

6、采用哈希表存储元素 7、4、5、3、6、2、8、9，设哈希函数为 $H(\text{key}) = (\text{key}^2 + 2) \text{MOD } 9$ ，采用链地址法处理冲突，

(1) 请画出依次存储上述元素后该哈希表的存储结构，要求每个链表中元素从小至大排列；

(2) 假定每个元素的查找概率相等，请计算平均成功查找长度。

7、设关键字序列为 (4, 28, 15, 37, 76, 23, 57, 19, 45, 6)，请写出希尔排序的过程。要求按从小到大的次序排列，设增量序列为 5、3、1。

8、从二叉树深度的定义可知，二叉树的深度应为其左、右子树深度的最大值加 1，请据此补充完整以下求解二叉树深度的函数 TreeDepth:

```
int TreeDepth (BiTree T) {
    if (!T) depth=0;
    else {
        depthLeft=TreeDepth(T->lchild);
        depthRight=TreeDepth(T->rchild);
        depth=1+(depthLeft>depthRight?depthLeft:depthRight);
    }
    return depth;
}
```

(1) 已知二叉树采用二叉链表存储，二叉链表的存储结构为:

typedef struct BiTNode {

int data;

struct BiTNode *lchild, *rchild;

} BiTNode, *BiTree;

(2) 使用教材中的类 C 代码描述即可。

(3) 要求 TreeDepth 为递归函数。

$$\frac{C_2^n}{n+1}$$

Handwritten solutions for the problems above:

Problem 6: Hash table using chain address method. The hash function is $H(\text{key}) = (\text{key}^2 + 2) \text{MOD } 9$. The elements are 7, 4, 5, 3, 6, 2, 8, 9. The hash values are calculated as follows:

- 7: $7^2 + 2 = 49 + 2 = 51 \text{MOD } 9 = 6$
- 4: $4^2 + 2 = 16 + 2 = 18 \text{MOD } 9 = 0$
- 5: $5^2 + 2 = 25 + 2 = 27 \text{MOD } 9 = 0$
- 3: $3^2 + 2 = 9 + 2 = 11 \text{MOD } 9 = 2$
- 6: $6^2 + 2 = 36 + 2 = 38 \text{MOD } 9 = 2$
- 2: $2^2 + 2 = 4 + 2 = 6 \text{MOD } 9 = 6$
- 8: $8^2 + 2 = 64 + 2 = 66 \text{MOD } 9 = 3$
- 9: $9^2 + 2 = 81 + 2 = 83 \text{MOD } 9 = 2$

The hash table structure is shown as a linked list:

```

0: [4] -> [5]
2: [3] -> [6]
6: [7] -> [2]

```

Problem 7: Shell sort process. The sequence is (4, 28, 15, 37, 76, 23, 57, 19, 45, 6). The increments are 5, 3, 1.

Initial sequence: 4, 28, 15, 37, 76, 23, 57, 19, 45, 6

Increment 5: 4, 28, 15, 37, 76, 23, 57, 19, 45, 6

Increment 3: 4, 28, 15, 37, 76, 23, 57, 19, 45, 6

Increment 1: 4, 6, 15, 37, 19, 23, 57, 28, 45, 76

Problem 8: Binary tree depth function. The function is defined as:

```

int TreeDepth (BiTree T) {
    if (!T) return 0;
    int depthLeft = TreeDepth(T->lchild);
    int depthRight = TreeDepth(T->rchild);
    return 1 + (depthLeft > depthRight ? depthLeft : depthRight);
}

```

Problem 9: Handwritten solutions for the problems above. The solutions include a hash table, a shell sort process, and a binary tree depth function.

日期: /

—

- 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关系和运算筹的学科
- 2.数据类型是一个值的集合和定义在此集合上的一组操作的总称
- 3.在问题规模 n 趋于无穷大时算法时间复杂度 $T(n)$ 的渐进上界，即函数 $T(n)$ 的数量级阶
- 4.栈是一种特殊的线性表，它是限定仅在表尾进行插入和删除的线性表。我们把允许插入和删除的一端称为栈顶，把另一端称为栈底，不含任何元素的栈称为空栈，栈又称为后进先出的线性表
- 5.哈夫曼树又称为最优二叉树。哈夫曼树是指具有相同节点的树中，加权路径长度最小的二叉树
- 6.图是由顶点的有穷非空集合和顶点之间的边的集合组成，通常表示为: $G=(V,E)$ ， V 是顶点集合， E 是边的集合

二

- 1.算法是对特定问题求解步骤的一种描述，它是指令的有限序列。有限性，确定性，可行性，输入性，输出性
- 2.集合中必定存在一个唯一的一个“第一个元素”；集合中必定存在唯一的一个“最后的元素”；除了最后元素之外，其他数据元素均有唯一的“后继”；除了第一个元素之外，其他元素均有唯一的“前驱”
- 3.头指针是指向链表中的第一个节点（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，数据域内可放空表标志或表长等信息；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
- 4.双亲表示法（以一组连续的存储空间存放树的结点，每个结点中附设一个指针指示其双亲结点在这连续的存储空间中的位置），孩子表示法（把每个结点的孩子排列起来，看成一个线性表，以单链表存储；令其头指针和结点的元素构成一个结点，并将所有这样的结点存放在一个地址连续的存储空间里），孩子兄弟表示法（用二叉链表作树的存储结构，链表中每个结点的两个指针域分别指向其第一个孩子结点和下一个兄弟结点）