

浅谈数据结构的学习方法

文 龙

四川大学锦城学院 计算机与软件学院 四川 成都 611731

【摘要】数据结构是指相互之间存在着一种或多种关系的数据元素的集合和该集合中数据元素之间的关系组成，而数据结构是关于编程中至关重要的一个部分，数据结构是教你如何写出更好的算法，而一个好的程序是算法 + 数据结构体现出来的，所以学习数据结构的方法极为重要，数据结构的分类又特别的多，极其容易混淆。

【关键词】线性结构；非线性结构；单链表；树

现如今，互联网时代发展迅速，在 21 世纪程序员一职尤为重要，并且存在大量的潜力，市场上大量的需求，以及科技的发展、程序的进步等等，而程序员最重要的就是要学会编程，而一个好的程序是需要数据结构 + 算法一起体现出来的，而一个程序得好与坏是可以提供数据结构中某些条件得出来，并且进行修改，让程序变得更好，更快，更好得操作，那么数据结构如此的重要是否人人都可以学好？是否人人可以掌握呢？那么又如何让自己可以更容易掌握学习方法呢？

1 数据结构的组成

1.1 有关的几个概念

- 1) 数据：凡是可输入计算机并能被计算机程序处理的符号总称。
- 2) 数据元素：数据的基本单位又称为记录或结点。
- 3) 数据结构：彼此之间存在一种或多种特定关系的数据元素集合。

1.2 数据结构的组成

[1] 数据结构分为三个部分：逻辑结构、物理结构、运算。

1.2.1 逻辑结构

它是不依赖计算机而存在的，它的含义是描述数据元素之间的某种逻辑关系，通常用一个二元组构成，如 $A=(D, R)$ ， $D=\{A, B, C, D, E, F, H\}$ ， $R=\{(D, A), (D, G), (D, A), (A, C), (G, E), (G, H), (E, F)\}$ 。

1.2.2 物理结构

物理结构它是需要计算机的，并且它是指数据的逻辑在计算机存储上的表现。

具体方法有：顺序法，链接法，哈希法（散列法）

以下举例常用的两个方法：顺序法和链接法

顺序法是指逻辑上相邻的数据元素，物理位置上也相邻，这种方法的好处是可以随机访问，但是却不好插入与删除。

链接法是指把结构体每个单元一分为二，一部分指指针域指存储下一个元素的地址，另一部分存储元素的数据信息^[1]。

1.2.3 算法及评价

算法：指解决某一特定类型问题的有限运算序列

（解决问题的方法思路）。

算法与程序的区别：算法需要进行加工之后才可以变成程序，而程序是可以直接运行的，但是算法不行，并且算法通常描述成函数。

算法的评价标准：①可读性，②健壮性（对用户输入的内容进行有效的判断），③正确性，④时间复杂度和空间复杂度（高效性）。

时间复杂度：一个算法种，基本语句重新执行的次数的数量级，通常用 $T(n)=O\{f(n)\}$ 表示。

如：for (i=1; i<=n; i++)

X++;

$f(n)=n$, $T(n)=O(n)$ ，这就是线性阶

又如：for (i=1; i<=n; i++)

for (j=1; j<=n; j++)

X++;

$f(n)=n*n$, $T(n)=O(n*n)$

2 线性结构与非线性结构（举例说明）

2.1.1 线性表

$(a_1, a_2, a_3 \dots a_i \dots a_n)$

1) 线性表的的含义：由 n 个相同性质的数据元素构成有限序列。

2) 线性表的表示：① n 表示线性表的长度，当 $n=0$ 时，该表为空表，② i 表示该数据元素在线性表的位置。

3) 线性表种元素之间的关系：依次相邻。

2.1.2 线性表的物理结构

1) 顺序存储的线性表称（顺序表）实现它需要要数组。

2) 优缺点：①随机访问，速度快，②插入与删除都不方便（如大量元素的移动等）。

2.1.3 算法的实现

$(a_1, a_2, a_3 \dots a_{k-1}, a_k, a_{k+1} \dots a_n)$

赋值法：令 $a_1=9, a_2=2, a_3=6, a_4=5, a_5=7$ ，令 $K=3$

则变成 $a_1=9, a_2=2, a_3=5, a_4=7$

```
代码:  
Delete (int a[],int n, int k)  
Int i;  
If(k<1||k>n)  
Printf(“删除失败”) // 要删除的元素不在这  
个范围内  
else  
{  
for(i=k;i<n;i++)  
a[i-1]=a[i];  
}  
n--;
```

2.2.1 树和二叉树

树的定义：由 n 个结点构成的有限集。
其中存在唯一一个称为根的结点

2.2.2 树的相关概念

结点：根，孩子，双亲，子孙，兄弟，叶子等。

结点的度：该结点的子孙个数就叫度。

树的度：树中所有度的最大值

结点的层次：根为第一层，根的孩子为第二层，

依此类推。

根的深度：树中结点层次的最大值。

2.2.3 二叉树（最大度不超过 2）

定义：树中每个结点最多只有两颗子树的树。

特点：①每个结点的度最大不超过 2，②子树：有左右之分，顺序不能颠倒。

3 学习方法

3.1 理解知识为基础，刷题为辅助

学习好数据结构首先就要把基础知识整扎实，基础知识是最为重要的，从读书这么多年以来，没有一颗学科是不学好基础知识就可以学好的，并且当你学好基础知识时，刷题时可以提高效率以及正确率，刚开始学习时，一定要把判断一个程序的好坏的标准理解清楚，像时间复杂度和空间复杂度这两个就是判断一个程序好坏的标准，前期学习时可以多刷这个环节的知识。

当学习到线性结构时，如线性表，这时候就需要进行以下几个步骤：①画图，②把一个大操作分成若干小操作，③判断清楚每个小部分用什么方法，工具，④写代码的框架，⑤验证。

其中画图是用来帮助自己理解题中的内容，把大

操作分成若干小操作可以更好的分步进行，更快的写出框架出来，最重要的是最后一步的验证，这是很多同学写完之后不会进行的一个步骤，这个步骤其实是极其重要的，有可能一个小条件就会导致整个算法失败，那么到最后编程就不会实现。

掌握每一种类型的表示方式与思路

数据结构大的类型一共分为线性结构与非线性结构，其中线性结构的我举例说明了线性表，学习线性表插入时，在一个线性表中插入，先考虑两种类型的表哪一种比较容易好实现这个操作，那么我们就想到了链式存储方式，链式存储方式种，我们需要用到结构体与指针，所以我们需要有这两方面的知识点，当我们建立起了链式存储表时，我们可以按位置插入，也可以按照顺序插入，其中按照有序插入的话，我们就需要引用两个指针变量。

代码如下：

```
Insert (NODE *p, int x)  
{  
NODE *P=head,*S;  
While (p->next&&P->next->data<x)  
p=p->next;  
S=(NODE*)malloc(sizeof(NODE));  
>data=x;  
>next=p->next;  
P->next=s;  
}
```

结语

以上就是我对数据结构的心得，数据结构对于程序员来说是极其重要的，它不仅仅是算法而已，它还体现出一个程序的“灵魂”，一个程序到底是如何被编译出来的，它会完全见证，因此学习数据结构需要不断的总结，刷题，得出属于自己的东西来。

【参考文献】

[1] 严蔚敏 李冬梅 吴伟民《数据结构（c 语言版 | 第二版）》