

什么是广义表、广义表及定义详解

前面讲过，[数组](#)即可以存储不可再分的数据元素（如数字 5、字符 'a'），也可以继续存储数组（即 n 维数组）。但需要注意的是，以上两种数据存储形式绝不会出现在同一个数组中。例如，我们可以创建一个整形数组去存储 {1,2,3}，我们也可以创建一个二维整形数组去存储 {{1,2,3},{4,5,6}}，但数组不适合用来存储类似 {1,{1,2,3}} 这样的数据。

有人可能会说，创建一个二维数组来存储{1,{1,2,3}}。在存储上确实可以实现，但无疑会造成存储空间的浪费。

对于存储 {1,{1,2,3}} 这样的数据，更适合用[广义表](#)结构来存储。

什么是广义表

[广义表](#)，又称[列表](#)，也是一种线性存储结构。

同数组类似，广义表中既可以存储不可再分的元素，也可以存储广义表，记作：

$$LS = (a_1, a_2, \dots, a_n)$$

其中，LS 代表广义表的名称， a_n 表示广义表存储的数据。广义表中每个 a_i 既可以代表单个元素，也可以代表另一个广义表。

原子和子表

通常，广义表中存储的单个元素称为 "[原子](#)"，而存储的广义表称为 "[子表](#)"。

例如创建一个广义表 $LS = \{1, \{1, 2, 3\}\}$ ，我们可以这样解释此广义表的构成：广义表 LS 存储了一个原子 1 和子表 {1,2,3}。

以下是广义表存储数据的一些常用形式：

- $A = ()$ ：A 表示一个广义表，只不过表是空的。
- $B = (e)$ ：广义表 B 中只有一个原子 e。
- $C = (a, (b, c, d))$ ：广义表 C 中有两个元素，原子 a 和子表 (b, c, d)。
- $D = (A, B, C)$ ：广义表 D 中存有 3 个子表，分别是A、B和C。这种表示方式等同于 $D = ((), (e), (b, c, d))$ 。
- $E = (a, E)$ ：广义表 E 中有两个元素，原子 a 和它本身。这是一个递归广义表，等同于： $E = (a, (a, (a, \dots)))$ 。

注意， $A = ()$ 和 $A = (())$ 是不一样的。前者是空表，而后者是包含一个子表的广义表，只不过这个子表是空表。

广义表的表头和表尾

当广义表不是空表时，称第一个数据（原子或子表）为"表头"，剩下的数据构成的新广义表为"表尾"。

强调一下，除非广义表为空表，否则广义表一定具有表头和表尾，且广义表的表尾一定是一个广义表。

例如在广义表中 $LS = \{1, \{1, 2, 3\}, 5\}$ 中，表头为原子 1，表尾为子表 $\{1, 2, 3\}$ 和原子 5 构成的广义表，即 $\{\{1, 2, 3\}, 5\}$ 。

再比如，在广义表 $LS = \{1\}$ 中，表头为原子 1，但由于广义表中无表尾元素，因此该表的表尾是一个空表，用 $\{\}$ 表示。