

# 数据结构和算法的关系和区别

由于大量数据结构教程中都将数据结构的知识和算法掺杂起来讲，使很多初学者认为数据结构就是在讲算法，这样理解是不准确的。

数据结构和算法之间完全是两个相互独立的学科，如果非说它们有关系，那也只是互利共赢、“ $1+1>2$ ”的关系。

最明显的例子，如果你认为数据结构是在讲算法，那么大学我们还学《算法导论》，后者几乎囊括了前者使用的全部算法，有什么必要同时开设这两门课程呢？

我们还可以从分析问题的角度去理清数据结构和算法之间的关系。通常，每个问题的解决都经过以下两个步骤：

- 1. 分析问题，从问题中提取出有价值的数据，将其存储；
- 2. 对存储的数据进行处理，最终得出问题的答案；

数据结构负责解决第一个问题，即数据的存储问题。通过前面的学习我们知道，针对数据不同的逻辑结构和物理结构，可以选出最优的数据存储结构来存储数据。

而剩下的第二个问题，属于算法的职责范围。算法，从表面意思来理解，即解决问题的方法。我们知道，评价一个算法的好坏，取决于在解决相同问题的前提下，哪种算法的效率最高，而这里的效率指的就是处理数据、分析数据的能力。

因此我们得出这样的结论，数据结构用于解决数据存储问题，而算法用于处理和分析数据，它们是完全不同的两类学科。

也正因为如此，你可以认为数据结构和算法存在“互利共赢、 $1+1>2$ ”的关系。在解决问题的过程中，数据结构要配合算法选择最优的存储结构来存储数据，而算法也要结合数据存储的特点，用最优的策略来分析并处理数据，由此可以最高效地解决问题。



图 1 顺序表存储数据示意图

例如，有这样一个问题，计算“ $1+2+3+4+5$ ”的值。这个问题我们可以这样来分析：

- 计算 1、2、3、4 和 5 的和，首先要选择一种数据存储方式将它们存储起来，通过前面的学习我们知道，数据之间具有“一对一”的逻辑关系，最适合用[线性表](#)来存储。结合算法的实现，我们选择顺序表来存储数据（而不是[链表](#)），如图 1 所示；
- 接下来，我们选择算法。由于数据集中存放，因此我们可以设计这样一个算法，使用一个初始值为 0 的变量 num 依次同存储的数据做“加”运算，最后得到的新 num 值就是最终结果。

选择顺序表而不是链表的原因，是顺序表遍历数据比链表更高效。后续讲顺序表时会做详细介绍。

**联系方式    购买教程（带答疑）**