

2020/05/29_数据结构_第1课_时间复杂度、动态数组

笔记本： 数据结构

创建时间： 2020/5/29 星期五 15:17

作者： ileemi

标签： 动态数组, 时间复杂度

- [课程介绍](#)
- [时间复杂度（算法的时间复杂度）](#)
 - [算法 VS 程序](#)
 - [好算法的标准](#)
 - [算法复杂度的计算](#)
 - [时间复杂度的表示](#)
 - [常见的时间复杂度](#)
 - [拿时间换空间，拿空间换时间](#)
- [动态数组](#)

课程介绍

数据结构 --> 解决具有复杂关系的大量数据的存储问题

数据结构大致包含以下几种存储结构：

- 线性表：顺序表、链表、栈、队列
- 树结构：普通树、二叉树、线索二叉树
- 图存储结构

顺序表：顺序存储

链表：随用随申请，数据的存储位置在内存中不是连续的，存储的位置是随机的

栈和队列：栈和队列隶属于线性表，是特殊的线性表

栈：栈中的元素只能从线性表的一端进出（另一端封死），且要遵循“先入后出”的原则，即先进栈的元素后出栈

队列：队列中的元素只能从线性表的一端进，从另一端出，且要遵循“先入先出”的特点，即先进队列的元素也要先出队列

树：适合存储“一对多”关系的数据

图：适合存储“多对多”关系的数据

时间复杂度（算法的时间复杂度）

算法：即解决问题的方法。同一个问题，使用不同的算法，虽然得到的结果相同，但是耗费的时间和资源是不同的。

算法 VS 程序

很多人误以为程序就是算法，其实不然：算法是解决某个问题的想法、思路；而程序是在心中有算法的前提下编写出来的可以运行的代码。

算法相当于是程序的雏形。当解决问题时，首先心中要有解决问题的算法，围绕算法编写出程序代码。

好算法的标准

准确性、健壮性、运行效率

运行效率体现在两个方面：

- 算法的运行时间 --> 时间复杂度
- 运行算法所需的内存空间的大小 --> 空间复杂度

好算法的标准就是：在符合算法本身的要求的基础上，使用算法编写的程序运行的时间短，运行过程中占用的内存空间少，就可以称这个算法是“好算法”。

算法复杂度的计算

估算算法时间复杂度

在学习C语言的时候讲过，程序由三种结构构成：顺序结构、分支结构和循环结构。顺序结构和分支结构中的每段代码只运行一次；循环结构中的代码的运行时间要看循环的次数。

由于是估算算法的时间复杂度，相比而言，循环结构对算法的执行时间影响更大。所以，算法的时间复杂度，**主要看算法中使用到的循环结构中代码循环的次数**（称为“**频度**”）。次数越少，**算法的时间复杂度越低**。

时间复杂度的表示

算法的时间复杂度的表示方式为：

$O(\text{频度})$

这种表示法称为大 O 记法

求一个算法的时间复杂度需要对其进行简化：

- 去掉运行时间中的所有加法常数，例如： $2n^3 + n + 1 \rightarrow n^3 + n$
- 只保留最高项，例如： $2n^3 + n \rightarrow n^3$
- 使最高项的系数保持为1，例如： $2n^3 \rightarrow n^3$

常见的时间复杂度

例子	时间复杂度	术语
9	$O(1)$	常数阶
$3n+4$	$O(n)$	线性阶
$3n^2+4n+5$	$O(n^2)$	平方阶
$3\log(2)n+4$	$O(\log n)$	对数阶
$2n+3n\log(2)n+14$	$O(n\log n)$	$n\log n$ 阶
n^3+2n^2+4n+6	$O(n^3)$	立方阶
2^n	$O(2^n)$	指数阶

常用时间复杂度所消耗的时间从小到大依次是：

$O(1) < O(\log n) < (n) < O(n\log n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$

折半查找就是典型的对数阶

拿时间换空间，拿空间换时间

算法的时间复杂度和空间复杂度是可以相互转化的。

谷歌浏览器相比于其他的浏览器，运行速度要快。是因为它占用了更多的内存空间，以空间换取了时间。

动态数组

封装一个数组支持增删改查

操作：创建、增加、修改、删除、查询

增加接口：

头部：Head

尾部：Tail

指定位置：Insert

删除接口：

头部

尾部

指定位置