

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



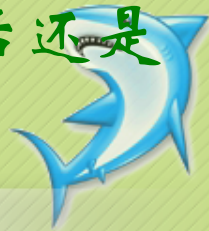
删除操作

- 所以删除算法的思路：
 - 如果删除位置不合理，抛出异常；
 - 取出删除元素；
 - 从删除元素位置开始遍历到最后一个元素位置，分别将它们都向前移动一个位置；
 - 表长-1。
- 实现代码：ListDelete.c



删除操作

- 现在我们分析一下，插入和删除的时间复杂度。
- 最好的情况：插入和删除操作刚好要求在最后一个位置操作，因为不需要移动任何元素，所以此时的时间复杂度为 $O(1)$ 。
- 最坏的情况：如果要插入和删除的位置是第一个元素，那就意味着要移动所有的元素向后或者向前，所以这个时间复杂度为 $O(n)$ 。
- 至于平均情况，就取中间值 $O((n-1)/2)$ 。
- 按照前边游戏秘籍指导，平均情况复杂度简化后还是 $O(n)$ 。



线性表顺序存储结构的优缺点

- 线性表的顺序存储结构，在存、读数据时，不管是哪个位置，时间复杂度都是 $O(1)$ 。而在插入或删除时，时间复杂度都是 $O(n)$ 。
- 这就说明，它比较适合元素个数比较稳定，不经常插入和删除元素，而更多的操作是存取数据的应用。
- 那我们接下来给大家简单总结下线性表的顺序存储结构的优缺点：



线性表顺序存储结构的优缺点

- 优点：
 - 无须为表示表中元素之间的逻辑关系而增加额外的存储空间。
 - 可以快速地存取表中任意位置的元素。
- 缺点：
 - 插入和删除操作需要移动大量元素。
 - 当线性表长度变化较大时，难以确定存储空间的容量。
 - 容易造成存储空间的“碎片”。



线性表的链式存储结构

- 前面我们讲的线性表的顺序存储结构，它最大的缺点就是插入和删除时需要移动大量元素，这显然就需要耗费时间。
- 那我们能不能针对这个缺陷或者说遗憾提出解决的方法呢？要解决这个问题，我们就得考虑一下导致这个问题的原因！
- 为什么当插入和删除时，就要移动大量的元素？
- 原因就在于相邻两元素的存储位置也具有邻居关系，它们在内存中的位置是紧挨着的，中间没有间隙，当然就无法快速插入和删除。



线性表的链式存储结构

- 经过叽叽呱呱的讨论之后，我们派出几个童鞋跟大家分享一下思路。
- A童鞋：让当中每个元素之间都留有一个空位置，这样要插入一个元素时，就不至于要移动了。可一个空位置如何解决多个相同位置插入数据的问题呢？所以这个想法显然不行。
- B童鞋：那就让当中每个元素之间都留足够多的位置，根据实际情况制定空隙大小，比如每个元素间留10个空位。那么问题就显而易见了，造成资源的极大浪费，并且在同一个位置插入11次也不是不可能。



线性表的链式存储结构

- C童鞋：反正要在相邻元素间留多少空间都是有可能不够的，那不如干脆不要考虑相邻位置这个问题了。哪里有空位就放在哪里，记得在小甲鱼老湿的《零基础入门学习C语言》中讲到的指针刚好可以派上用场。每个元素多用一个位置来存放指向下一个元素的位置的指针。这样子从第一个元素可以找到第二个元素，第二个元素可以找到第三个元素，依此类推，所有的元素我们就都可以通过遍历而找到了。
- 好，太棒了，这个想法灰常好！掌声鼓励！



线性表链式存储结构定义

- 哎呀，又要讲定义的时候了，往往啊，这个时候就是搬出概念砸脚的时候了。我们不能这样丫，我们的课堂应该是欢乐与傻笑并存，智慧与邪恶同在。
- 好，那我们从爱情动作片开始吧~
- 这个无论是爱情片还是动作片，我们都感觉它没有悬疑片那么吸引人。。。。。
- 而悬疑片正是像链式存储那样一环套这一环，时而让你紧张，时而让你鸡冻！



线性表链式存储结构定义

- 线性表的链式存储结构的特点是用一组任意的存储单元存储线性表的数据元素，这组存储单元可以存在内存中未被占用的任意位置。
- 比起顺序存储结构每个数据元素只需要存储一个位置就可以了。现在链式存储结构中，除了要存储数据元素信息外，还要存储它的后继元素的存储地址（指针）。
- 也就是说除了存储其本身的信息外，还需存储一个指示其直接后继的存储位置的信息。

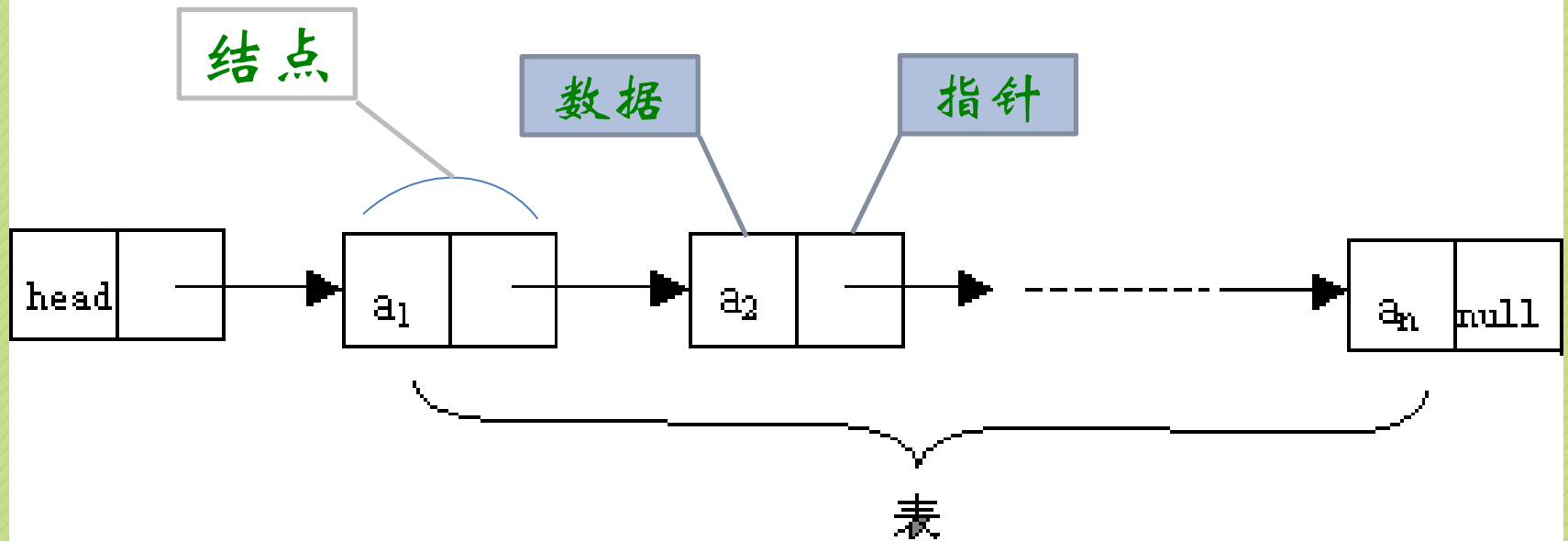


线性表链式存储结构定义

- 我们把存储数据元素信息的域称为数据域，把存储直接后继位置的域称为指针域。指针域中存储的信息称为指针或链。这两部分信息组成数据元素称为存储映像，称为结点(Node)。
- n 个结点链接成一个链表，即为线性表($a_1, a_2, a_3, \dots, a_n$)的链式存储结构。
- 因为此链表的每个结点中只包含一个指针域，所以叫做单链表。
- No pic you say a J8...



单链表



单链表

- 对于线性表来说，总得有个头有个尾，链表也不例外。我们把链表中的第一个结点的存储位置叫做头指针，最后一个结点指针为空(NULL)。

