

如何插入???

目录为空，请插入标题（字体左侧）后点击刷新按钮

一、无序链表

1. 节点：数据项+next (类似于指针的作用)
head-节点；(上一个节点的)next-下一个节点
- 2.

❖ 以整数数据项为例，(17, 26, 31, 54, 77, 93)的链表形式如图



3. 几种方法的集合：

add()
length()
search()
remove()

二、有序链表

0. 只是大小的排序，但是存储的位置一样时无序
1. 方法的改变：
 - (1) add()

一、定义：

链表是数据结构之一，其中的数据呈线性排列。在链表中，数据的添加和删除都较为方便，就是访问比较耗费时间。

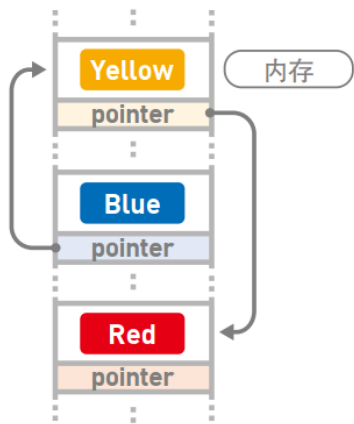
二、概念图：

01



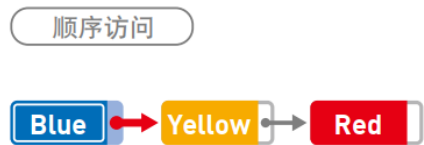
这就是链表的概念图。Blue、Yellow、Red 这 3 个字符串作为数据被存储于链表中。每个数据都有 1 个“指针”，它指向下一个数据的内存地址。

02

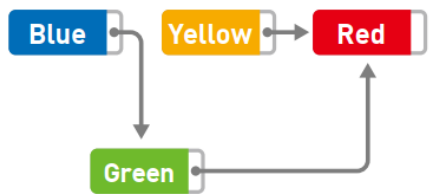


在链表中，数据一般都是分散存储于内存中的，无须存储在连续空间内。

03



因为数据都是分散存储的，所以如果想要访问数据，只能从第 1 个数据开始，顺着指针的指向一一往下访问（这便是顺序访问）。比如，想要找到 Red 这一数据，就得从 Blue 开始访问。



这时，只需要把 Green 指针指向的位置从 Yellow 变成 Red，删除就完成了。虽然 Yellow 本身还存储在内存中，但是不管从哪里都无法访问这个数据，所以也就没有特意去删除它的必要了。今后需要用到 Yellow 所在的存储空间时，只要用新数据覆盖掉就可以了。

三、运行时间

访问数据时，我们需要从链表头部开始查找（线性查找），如果目标数据在链表最后

的话，需要的时间就是 $O(n)$ 。

四、补充：

1. 这便是“循环链表”，也叫“环形链表”。

循环链表没有头和尾的概念。想要保存数量固定的最新数据时通常会使用这种链表。

循环链表



1. “双向链表”。使用这种链表，不仅可以从前往后，还可以从后往前遍历数据，十分方便。

但是，双向链表存在两个缺点：一是指针数的增加会导致存储空间需求增加；二是添加和删除数据时需要改变更多指针的指向。

双向链表

