# 1.2 链表的读取与查找

## 链表元素打印

*// 链表中的元素字符串形式*

**public** String **toString**() {

**if** (**this**.root == **null**) {

**return** "";

}

StringBuilder str = **new** StringBuilder();

Node node = **this**.root;

**while** (node != **null**) {

str.append(node.getContent()).append(" ");

node = node.getNext();

}

**return** str.toString();

}

***System.out.println(linkedList.toString());***

## 结点插入

头部插入：

Node的 next 指向 root，Node设置为 root。

尾部插入：

遍历到链表尾部，设置next 为 node。

中间插入：

遍历到插入位置，定位到root以及root.next，先断开后连接。

## 链表删除

头部删除：

Root指向第二个结点，原始root结点的next设置为 null

中间删除：

待删除结点的前置结点的next指向待删除结点的后向结点，待删除结点的next指向null

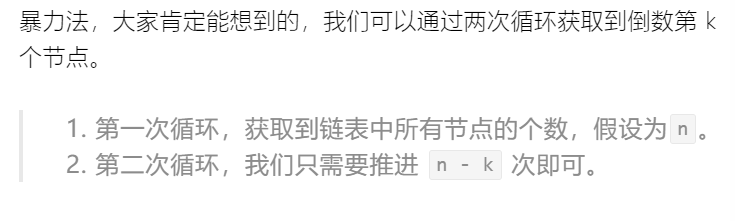
尾部删除：

待删除结点的前置结点的next设置为null

## 链表经典算法题——多指针法

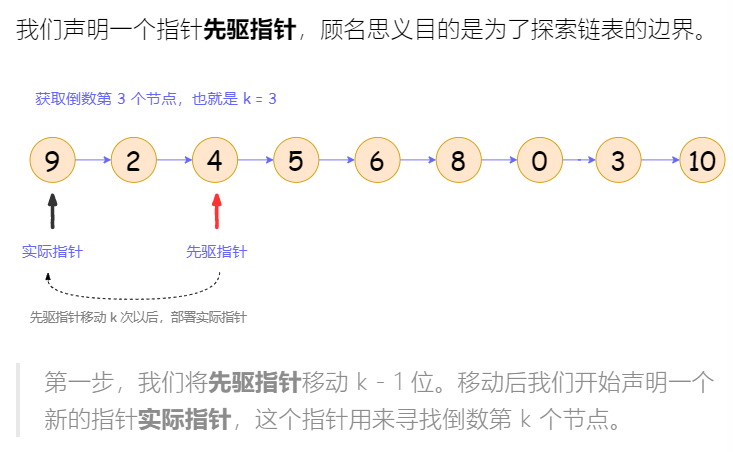
如何查找单向链表中倒数第k个结点？

暴力法：

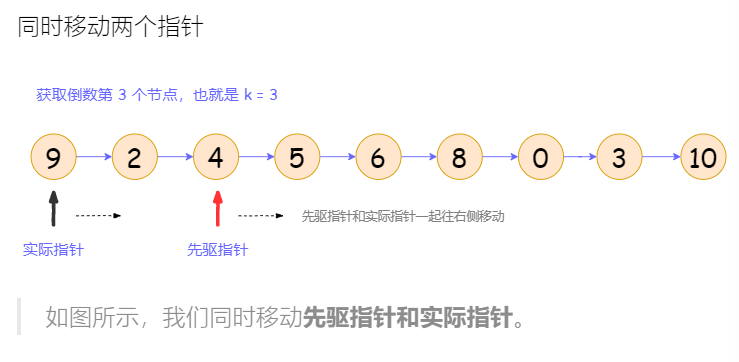


双指针法：

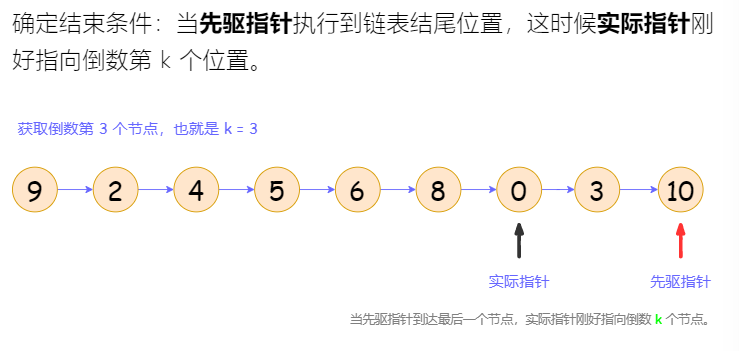
第一步，



第二步，



第三步，



多指针法寻找中间结点：

实际指针走一步，先驱指针走两步。

判断是否为有环链表： （过程类似于寻找中间结点）

