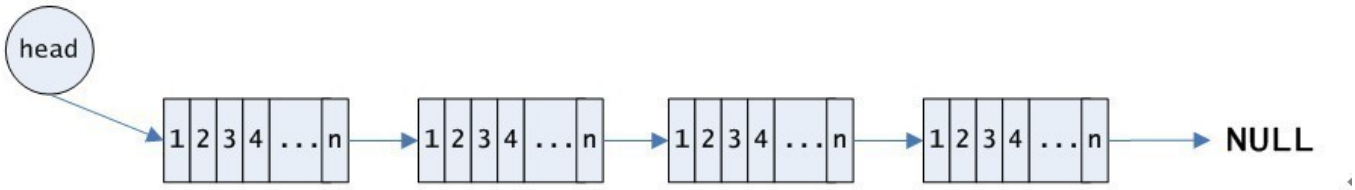
数据结构：块状链表

一、概述

有时候我们需要设计这样一种数据结构：它能快速在要求位置插入或者删除一段数据。先考虑两种简单的数据结构：数组和链表。数组的优点是
能够在O(1)的时间内找到所要执行操作的位
置，但其缺点是
无论是插入或删除都要移动
之后的所有数据，复杂度是O(n)的。链表优点是
能够在O(1)的时间内插入和删除一段数据，但缺点是
在寻找操作位置时，却要遍历整个链表，复杂度同样时O(n)的。这两种数据结构各有优缺点，我们可以把数组和链表的优点
结合来，这就构成了一个新的数据结构：块状链表
，结合数组和链表的优缺点的块状链表其各种操作的时间复杂度均为O(sqrt(n))。



二、块状链表的各种操作

块状链表是结合了数组和链表的优缺点，块状链表本身是一个链表，但是链表储存的并不是一般的数据，而是由这些数据组成的顺序表。每一个块状链表的节点，也就是顺序表，可以被叫做一个块。块状链表通过使用可变的顺序表的长度和特殊的插入、删除方式，可以在达到的复杂度。块状链表另一个特点是相对于普通链表来说节省内存，因为不用保存指向每一个数据节点的指针。

1、定位操作

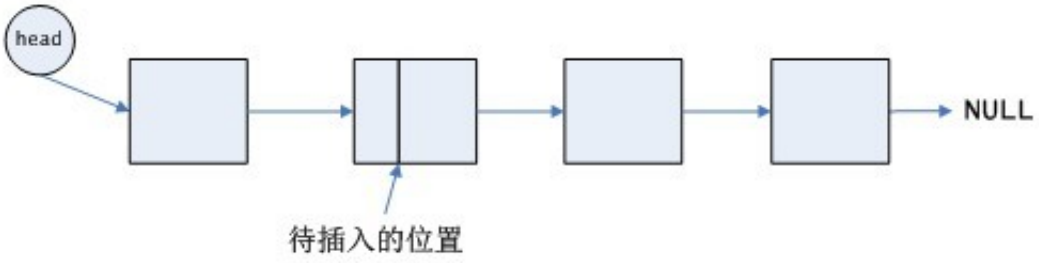
定位操作其实可以当作是查找，我们当然是先要定位到元素所在的节点，然后在该节点里面的数组里面寻找我们要的节点；

2、分裂节点

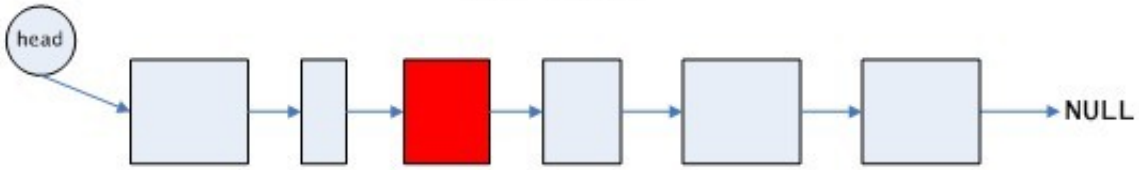
将某个节点分裂成两个节点

3、插入操作

首先定位要插入的位置，然后将所在节点分裂成两个节点，并将数据放到第一个节点的末尾。 如果要插入的是一大块数据，首先要将数据切成多个块其中、每个块对应一个块状链表的一个节点，并将这些块链起来，然后将它们插入那两个节点之间。插入的操作图类似下面：



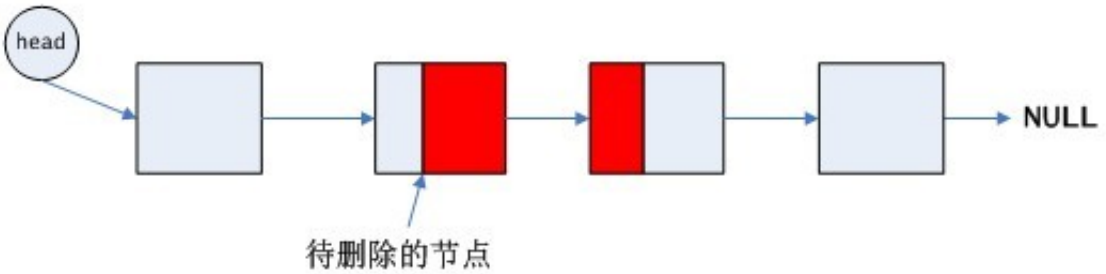
插入之前



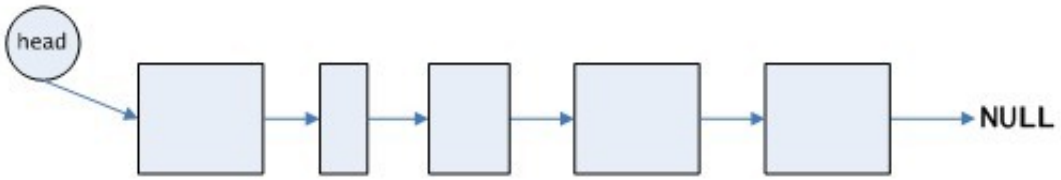
插入之后

4、删除操作

首先定位删除元素的位置，然后按照数组删除元素的方法删除该数据。如果删除一大块数据，首先要定位数据块首元素和末元素所在的位置，然后分别将它们所在的节点分裂成两个节点，最后删除首元素和末元素之间的节点即可。



删除之前



删除之后

三、关键技术

从总体上来看，维护一个链表，链表中的每个单元中包含一段数组，以及这个数组中的数据个数。每个链表中的数据连起来就是整个数据。设链表长度为a，每个单元中的数组长度是b。无论是插入或删除，在寻址时要遍历整个链表，复杂度是O(a)；对于插入操作，直接在链表中加入一个新单元，复杂度是O(1)；对于删除操作，会涉及多个连续的单元，如果一个单元中的所有数据均要删除，直接删除这个单元，复杂度是O(1)，如果只删除部分数据，则要移动数组中的数据，复杂度是O(b)。总的复杂度是O(a+b)。因为ab=n，取a=b=√n，则总的复杂度是O(√n)。
问题是如何维护a和b大致等于√n？
在执行插入操作后，如果当前单元中的数据个数>2√n，则将当前单元分割成两个新单元，每个单元中的数据个数保持为2√n。在执行删除操作后，如果当前单元和当前单元的下一个单元的数据个数和<√n，则将两个单元合并成一个新单元。执行上述维护操作需要移动数组中的数据，复杂度是O(b)，对于单元的分割和合并均是O(1)的，总的复杂度是O(b)的。这样，维护操作并不会使总复杂度增加。最终得到一个复杂度是O(√n)的数据结构。

四、应用

1. 文本编辑器：http://download.noi.cn/T/noi/noi2003A.pdf
   另一种实现：http://www.byvoid.com/blog/noi-2003-editor/
2. 维护数列：http://download.noi.cn/T/noi/noi2005A.pdf
   另一种实现：http://www.byvoid.com/blog/noi-2005-sequence/

本文资料来自互联网，欢迎大家指点里面错误之处。