



数据结构与算法 (Python版)

无序表抽象数据类型及Python实现

陈斌 北京大学 gischen@pku.edu.cn

列表List：什么是列表？

- ❖ 在前面基本数据结构的讨论中，我们采用Python List实现了多种线性数据结构
 - ❖ 列表List是一种简单强大的数据集结构，提供了丰富的操作接口
- 但并不是所有的编程语言都提供了List数据类型，有时候需要程序员自己实现。

列表List: 什么是列表?

- ❖ 一种数据项按照**相对位置**存放的数据集
特别的, 被称为“无序表unordered list”
其中数据项只按照存放位置来索引, 如第1个、第2个……、最后一个等。
(为了简单起见, 假设表中不存在重复数据项)
- ❖ 如一个考试分数的集合 “54, 26, 93, 17, 77和31”
- ❖ 如果用无序表来表示, 就是[54, 26, 93, 17, 77, 31]

抽象数据类型：无序表List

❖ 无序表List的操作如下：

List()：创建一个空列表

add(item)：添加一个数据项到列表中，假设
item原先不存在于列表中

remove(item)：从列表中移除item，列表被修
改，item原先应存在于表中

search(item)：在列表中查找item，返回布尔
类型值

isEmpty()：返回列表是否为空

size()：返回列表包含了多少数据项

抽象数据类型：无序表List

❖ 无序表List的操作如下：

`append(item)`：添加一个数据项到表末尾，假设`item`原先不存在于列表中

`index(item)`：返回数据项在表中的位置

`insert(pos, item)`：将数据项插入到位置`pos`，假设`item`原先不存在与列表中，同时原列表具有足够多个数据项，能让`item`占据位置`pos`

`pop()`：从列表末尾移除数据项，假设原列表至少有1个数据项

`pop(pos)`：移除位置为`pos`的数据项，假设原列表存在位置`pos`

采用链表实现无序表

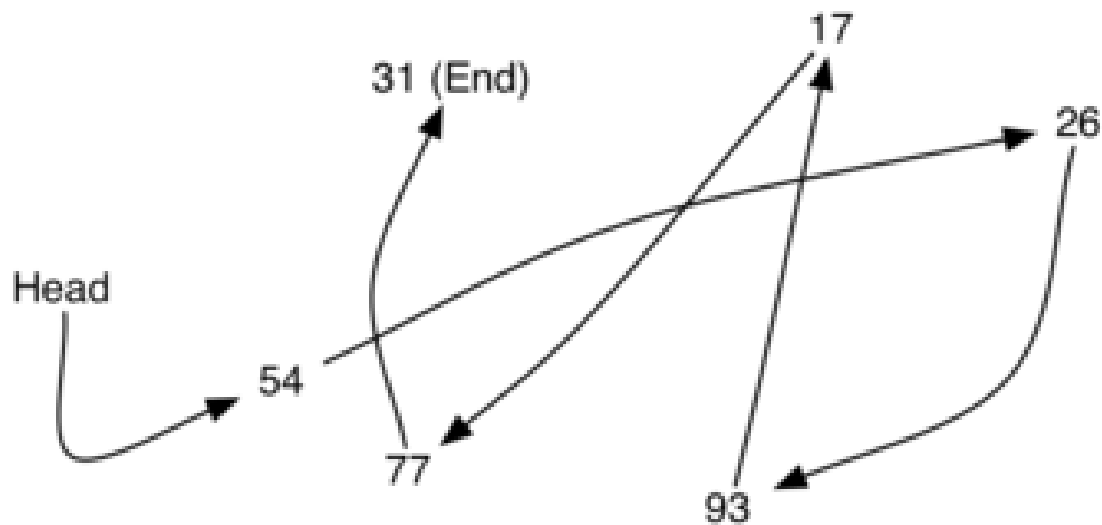
- ❖ 为了实现无序表数据结构，可以采用链接表的方案。
- ❖ 虽然列表数据结构要求保持数据项的前后相对位置，但这种前后位置的保持，并不要求数据项依次存放在连续的存储空间

31 17 26
54 77 93

采用链表实现无序表

❖ 如下图，数据项存放位置并没有规则，但如果在数据项之间建立**链接指向**，就可以保持其**前后相对位置**

第一个和最后一个数据项需要显式标记出来，一个是队首，一个是队尾，后面再无数据了。

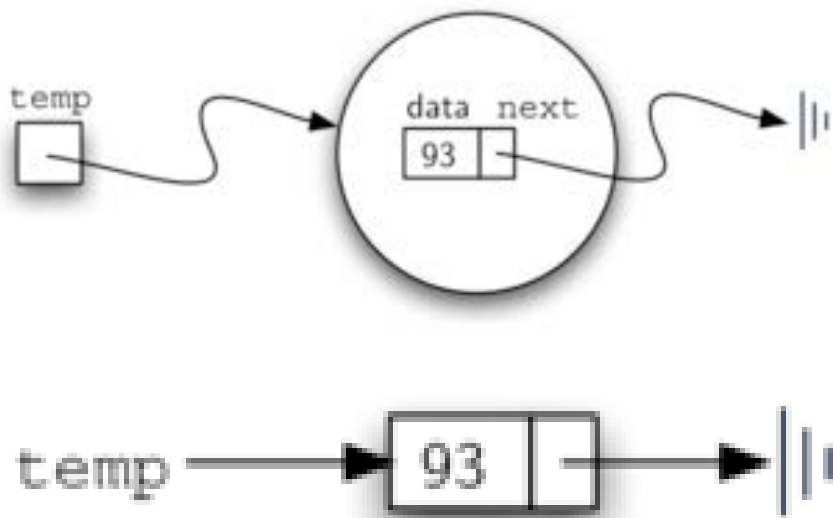


链表实现：节点Node

❖ 链表实现的最基本元素是节点Node

每个节点至少要包含2个信息：**数据项**本身，以及指向下一个节点的**引用**信息

注意next为None的意义是没有下一个节点了，这个很重要



链表实现：节点Node

```
class Node:
    def __init__(self, initdata):
        self.data = initdata
        self.next = None

    def getData(self):
        return self.data

    def getNext(self):
        return self.next

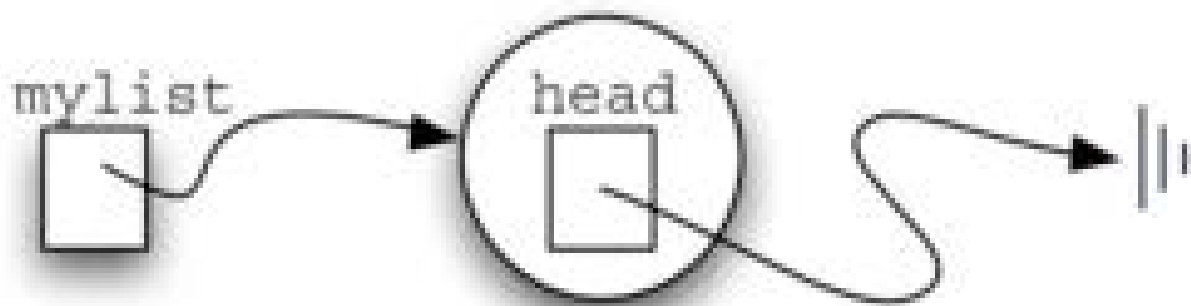
    def setData(self, newdata):
        self.data = newdata

    def setNext(self, newnext):
        self.next = newnext

>>> temp= Node(93)
>>> temp.getData()
93
>>> |
```

链表实现：无序表UnorderedList

- ❖ 可以采用**链接节点**的方式构建数据集来实现无序表
- ❖ 链表的**第一个和最后一个节点**最重要
如果想访问到链表中的所有节点，就必须从第一个节点开始沿着链接遍历下去



链表实现：无序表UnorderedList

❖ 所以无序表必须要有对**第一个节点**的引用信息

设立一个属性head，保存对第一个节点的引用

空表的head为None

```
class UnorderedList:

    def __init__(self):
        self.head = None

>>> mylist= UnorderedList()
>>> print mylist.head
None
```

链表实现：无序表UnorderedList

❖ 随着数据项的加入，无序表的head始终指向链条中的**第一个节点**

注意！无序表mylist对象本身并不包含数据项
(数据项在节点中)

其中包含的head只是对首个节点Node的引用

判断空表的isEmpty()很容易实现

- `return self.head == None`

