

选6、链表的概念与基本操作1—构成与遍历

1、链表的概念与特性

链表指的是将需要处理的数据对象以节点的形式，通过指针串联在一起的一种数据结构。

链表包括一个头指针和若干个链表节点，其中一个链表节点又包括数据区域和指针区域。

单向链表只有一个指针区域，指向下一个链表节点，双向链表有两个指针区域，分别指向上一个链表节点和下一个链表节点，循环链表的最后一个节点的指针区域指向头一个节点。

链表的特性包括：

- 1) 同一链表中每个节点的结构均相同
- 2) 每个链表必定有一个头指针（head），以实现链表的引用和边界处理
- 3) 链表占用的空间不固定

2、链表的创建

依然用Python的列表来模拟链表的数据结构。依然值得注意的是，在这里Python的列表被看作是计算机中一片连续的内存空间。

这里需要用到两个变量：列表data_a和整数型head_a，其中data_a用来存放链表的具体内容，head_a用来表示链表表头指针。

- 1) 创建空链表：

```
data_a=[]  
head_a=-1
```

- 2) 在链表插入20个随机降序的数据元素：

```
from random import randint  
data_a.append([randint(95,100),head_a]) #插入链表的第一个元素，具体数据为95-100的随机数，指针改为-1  
head_a=0  
for i in range(1,20):  
    tmp=data_a[i-1][0]-randint(1,5) #在上一节点数据的基础上，随机减去1-5产生新数据  
    data_a.append([tmp,data_a[i-1][1]]) #增加新的节点，指针设置为上一节点指针  
    data_a[i-1][1]=i #将上一节点指针设置为当前节点的索引  
head_a,data_a
```

```
(0,  
[[96, 1],  
[93, 2],  
[91, 3],  
[90, 4],  
[87, 5],  
[84, 6],  
[83, 7],  
[80, 8],  
[77, 9],  
[76, 10],  
[73, 11],  
[72, 12],  
[69, 13],  
[64, 14],  
[61, 15],  
[59, 16],  
[55, 17],  
[52, 18],  
[51, 19],  
[48, -1]])
```

可以看到，head_a=0，表示该链表的头号节点在列表data_a的0号位置上，而在data_a中，任何一个列表元素都是一个包含数据区域和指针区域的列表。其中指针区域指向的是后趋节点的索引。最后一个链表节点的指针为-1，表示这是最后一个链表节点。

3、链表的遍历访问

1) 链表遍历的一般过程：

设置访问指针ka，初始值为链表头指针

当ka不为-1时，不停的循环：

 输出当前访问到的链表元素数据

 将访问指针设置为当前链表元素的后趋节点指针

```
ka=head_a #设置访问指针ka，从链表头指针开始
while ka!=-1: #当访问指针不为-1时，不停循环
    print(data_a[ka][0]) #打印输出当前访问到的链表元素数据
    ka=data_a[ka][1] #通过当前链表元素的后趋指针调整访问指针
```

```
96
91
87
83
81
80
77
72
67
62
57
54
49
46
42
38
34
29
28
23
```

2) 为了方便后面使用，可以将链表遍历的过程写成函数形式：

```
def show_chain(data,head):
    k=head
    while k!=-1:
        print(data[k][0])
        k=data[k][1]
```