数据结构和算法

作者：小甲鱼

让编程改变世界

Change the world by program

头指针与头结点的异同

- 上
- 拿

存储

任

。

头指针与头结点的异同

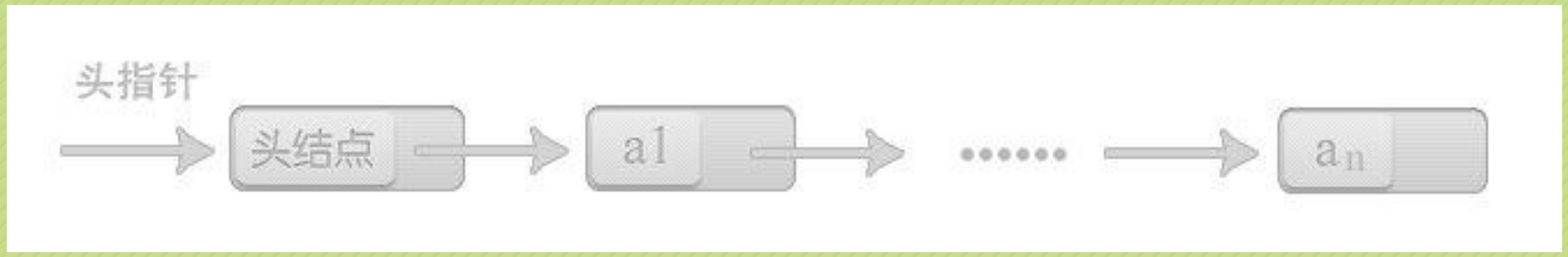
- 那有童鞋就疑惑了，既然头结点的数据域不存储任何信息，那么头指针和头结点又有何异同呢？
- 头指针
  - 头指针是指链表指向第一个结点的指针，若链表有头结点，则是指向头结点的指针。
  - 头指针具有标识作用，所以常用头指针冠以链表的名字（指针变量的名字）。
  - 无论链表是否为空，头指针均不为空。
  - 头指针是链表的必要元素。

头指针与头结点的异同

- 头结点
  - 头结点是为了操作的统一和方便而设立的，放在第一个元素的结点之前，其数据域一般无意义（但也可以用来存放链表的长度）。
  - 有了头结点，对在第一元素结点前插入结点和删除第一结点起操作与其它结点的操作就统一了。
  - 头结点不一定是链表的必须要素。
- No pic you say a J8…

单链表存储结构

- 单链表图例：



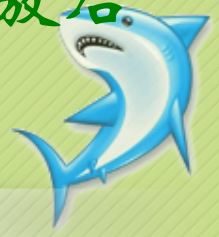
- 空链表图例：

单链表存储结构

- 我们在C语言中可以用结构指针来描述单链表。

```
typedef struct Node
{
    ElemType data;     // 数据域
    struct Node* Next; // 指针域
} Node;
typedef struct Node* LinkList;
```

- 我们看到结点由存放数据元素的数据域和存放后继结点地址的指针域组成。



单链表存储结构

- 假设p是指向线性表第i个元素的指针，则该结点ai的数据域我们可以用p->data的值是一个数据元素，结点ai的指针域可以用p->next来表示，p->next的值是一个指针。
- 那么p->next指向谁呢？当然指向第i+1个元素！也就是指向ai+1的指针。
- 问题：
  - 如果p->data =ai，那么p->next->data =?
- 答案：p->next->data =ai+1。

单链表的读取

- 在线性表的顺序存储结构中，我们要计算任意一个元素的存储位置是很容易的。
- 但在单链表中，由于第i个元素到底在哪？我们压根儿没办法一开始就知道，必须得从第一个结点开始挨个儿找。
- 因此，对于单链表实现获取第i个元素的数据的操作GetElem，在算法上相对要麻烦一些，大家不妨先思考一下。

单链表的读取

- 获得链表第i个数据的算法思路：
  - 声明一个结点p指向链表第一个结点，初始化j从1开始；
  - 当j<i时，就遍历链表，让p的指针向后移动，不断指向一下结点，j+1；
  - 若到链表末尾p为空，则说明第i个元素不存在；
  - 否则查找成功，返回结点p的数据。
- 有了以上的思路提示，小甲鱼邀请大家再度进行头脑风暴：算法的C语言实现代码，GetElem.c

单链表的读取

- 说白了，就是从头开始找，直到第i个元素为止。
- 由于这个算法的时间复杂度取决于i的位置，当i=1时，则不需要遍历，而i=n时则遍历n-1次才可以。因此最坏情况的时间复杂度为O(n)。
- 由于单链表的结构中没有定义表长，所以不能实现知道要循环多少次，因此也就不方便使用for来控制循环。
- 其核心思想叫做“工作指针后移”，这其实也是很多算法的常用技术。