

第二章 线性表上机实验

时间：第8周星期三（10月23日）

晚19:00-22:00

地点：基础实验大楼506

上机题目

□ 题目:

- 1. 顺序表的初始化, 建立, 插入, 查找, 删除。
- 2. 单链表的初始化, 建立, 插入, 查找, 删除。
- 3. 加分题: 采用循环链表实现约瑟夫问题;

□ 题目说明如下:

- 假定 **element type** 是 **int** 型;
- 设计方法可以任意转变 **element type** 类型 (建议采用 **typedef** 定义);
- 建立是指通过输入数据元素建立一个线性表, 建议用 **ctrl-z** 结束表示输入完毕 (也可以不用);
- 一般只要求做 **1** 和 **2**; 有能力的同学在做完 **1** 的基础上, 可以直接选择第 **3** 题, 此题为加分题。

上机题目

□ 课后：第10周星期日前提交上机报告

datastructurelife@163.com

□ 实验报告要求：

- 1. 实验题目的设计描述
- 2. 调试程序后得到的结果（截屏）
- 3. 源程序及程序运行结果打印清单（需要简单注释，说明函数功能、入口和出口参数）
- 4. 实验结论和结果分析（可选）

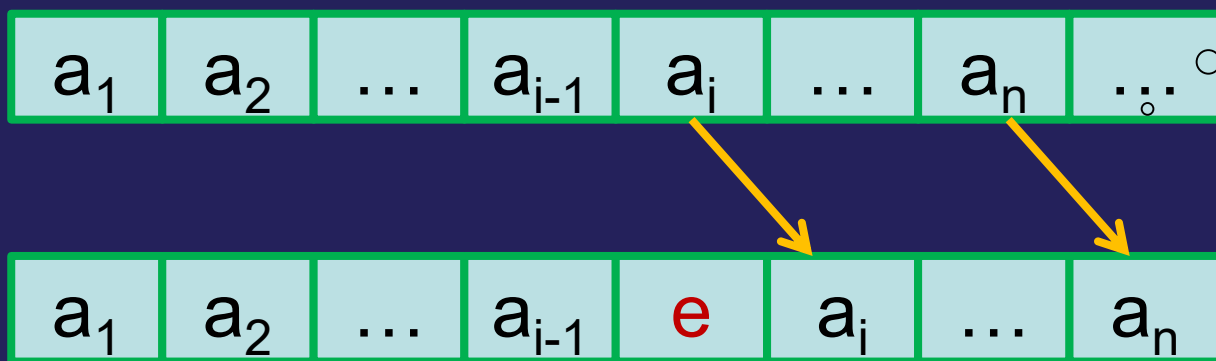


§ 2.2 线性表的顺序存储表示和实现

二、各个运算在顺序存储下的虚拟实现

ListInsert(&L, i, e) // 前插操作

考虑空间满的情况



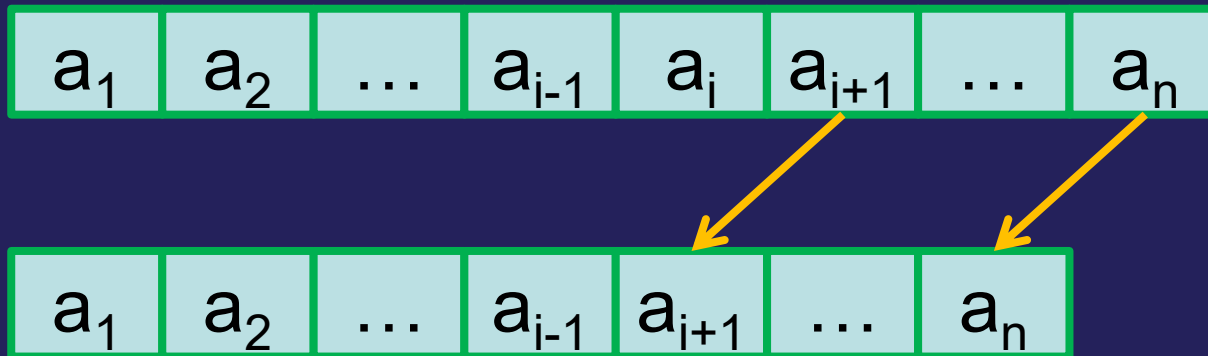
从 a_n 开始依次向后挪一位，一直到 a_i ；
将 e 填入原来 a_i 的位置，表的长度加。



§ 2.2 线性表的顺序存储表示和实现

二、各个运算在顺序存储下的虚拟实现

ListDelete(&L, i, &e) // 删除操作



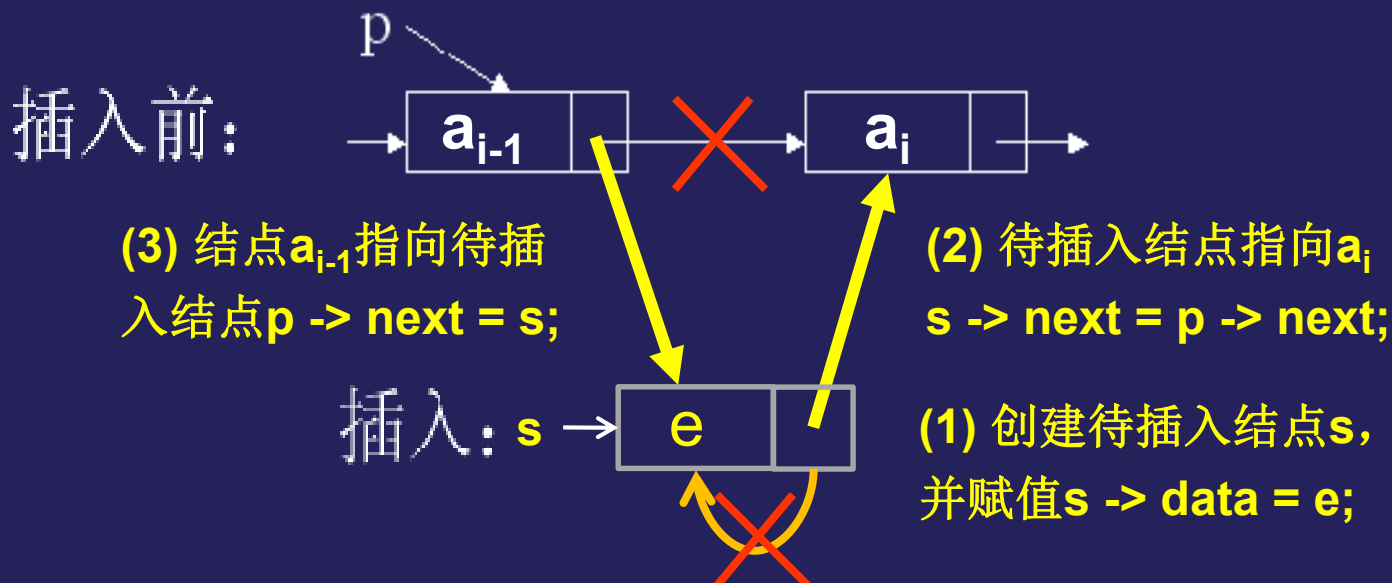
删除元素 a_i ，从 a_{i+1} 开始依次向前挪一位，一直到 a_n ；表的长度减1。



§ 2.3 线性表的链式存储表示和实现 单链表的操作实现

ListInsert(&L, i, e) // 插入数据元素

基本思路: 有序对 $\langle a_{i-1}, a_i \rangle$ 改变为 $\langle a_{i-1}, e \rangle$ 和 $\langle e, a_i \rangle$



思考: (2) 和 (3) 这两步是否可以颠倒? 为什么?



§ 2.3 线性表的链式存储表示和实现

单链表的操作实现

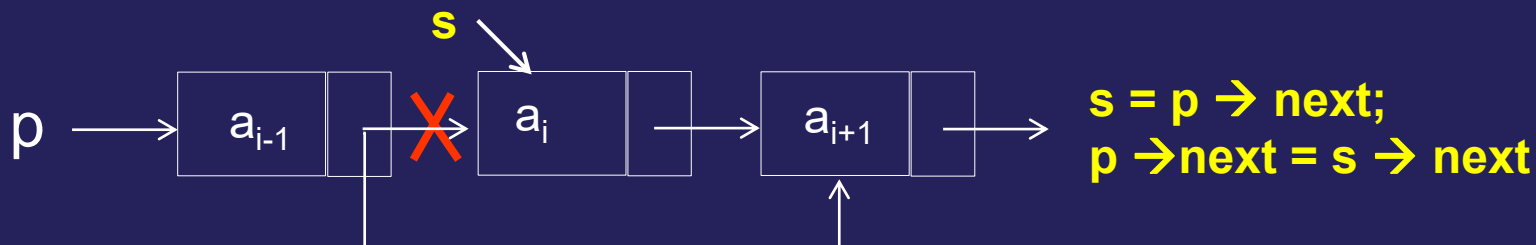
ListDelete(&L, i, e) // 删除数据元素

基本思路: 有序对 $\langle a_{i-1}, a_i \rangle$, $\langle a_i, a_{i+1} \rangle$ 改变为 $\langle a_{i-1}, a_{i+1} \rangle$

删除前



删除后



释放结点





§ 2.3 线性表的链式存储表示和实现 单链表的操作实现

建立单链表: `CreateList(&L, n)`

输入线性表元素，以单链式存储方式存储，创立单链表。链表是一个动态的结构，生成链表的过程是一个结点“逐个插入”的过程。



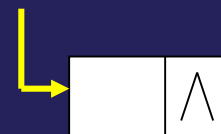
§ 2.3 线性表的链式存储表示和实现

单链表的操作实现

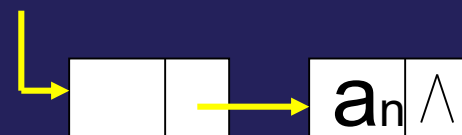
逆序法：从尾到头，即从最后一个元素结点逐个创建各个元素结点。每次都是插到当前链表的前面，即**头结点之后**。

操作步骤：

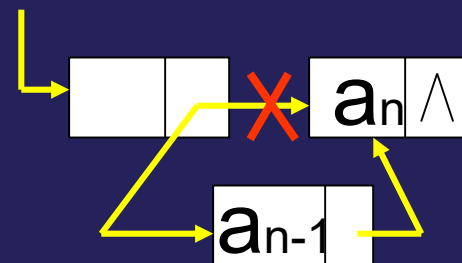
一、建立一个“空表”（头结点）；



二、输入数据元素 a_n ，建立结点并插入；



三、输入数据元素 a_{n-1} 建立结点并插入；



四、依次类推，直至输入 a_1 为止。



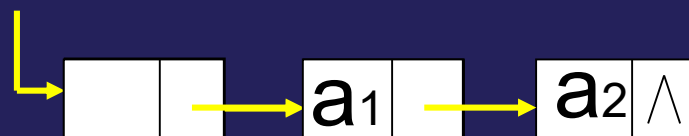
§ 2.3 线性表的链式存储表示和实现

单链表的操作实现

顺序法：从头到尾，即从第一个元素结点逐个创建各个元素结点。每次都是插入到当前链表的后面，即**表尾结点**。

操作步骤：

一、建立一个“空表”；



二、输入数据元素 a_1 ，建立结点并插入；

三、输入数据元素 a_2 建立结点并插入；

四、依次类推，直至输入 a_n 为止。