

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



线性表的抽象数据类型

- 上节课我们讲到了线性表的定义，讲到了所谓抽象数据类型就是把数据类型和相关操作捆绑在一起。
- 那么我们接下来分析一下，线性表应该有什么样的相关操作呢？
- 还是回到小甲鱼组织大家春游的例子，小甲鱼把鱼油们按照规律安排成一队，并且是长期使用这样的顺序排队，大家只需要记住自己的前驱鱼油就可以了。那么这个考虑和安排的过程其实就是一个线性表的创建和初始化过程。



线性表的抽象数据类型

- 一开始小甲鱼没经验呀，把鱼油们按照名字第一个字母的规律排队，发现排到最后有的高有的矮，导致队伍很难看，于是让鱼油们解散重新按照从矮到高排。
- 上边这么描述的过程其实就是线性表重置为空表的操作过程，接着给大家描述下删除数据。
- 排好队后，尼玛，发现黑夜童鞋由于昨晚吃太多麻辣烫，今天闹肚子来不了春游。那么由原来排在黑夜后边的小生开始往前挪。



线性表的抽象数据类型

- 当然有删除数据就有插入数据，黑夜童鞋下午肚子康复了，又回来了。黑夜童鞋说记得是排在斜阳后边的，所以让从斜阳后边的小生开始往后退一个位置，黑夜就可以插回自己的位置上了。
- 噢，我们活动开始了不久，隔壁来了春田花花旅游团都是美眉啊，各种调戏有木有？然后小甲鱼被她们的导游投诉了有木有？！导游说你们队伍里第八个那个家伙老是调戏我们队里的小花，他叫什么名字？小甲鱼查了下名单说，噢，他叫迷途。
- 好吧，这就是根据位序得到数据元素的例子。



线性表的抽象数据类型

- 好了，剩下的还有很多大家可以自己想象。
- 我们给大家总结下线性表的抽象数据类型定义：
- ADT 线性表 (List)
- Data

线性表的数据对象集合为 $\{a_1, a_2, \dots, a_n\}$ ，每个元素的类型均为DataType。其中，除第一个元素 a_1 外，每一个元素有且只有一个直接前驱元素，除了最后一个元素 a_n 外，每一个元素有且只有一个直接后继元素。数据元素之间的关系是一对一的关系。



线性表的抽象数据类型

- Operation

InitList(*L): 初始化操作, 建立一个空的线性表L。

ListEmpty(L): 判断线性表是否为空表, 若线性表为空, 返回true, 否则返回false。

ClearList(*L): 将线性表清空。

GetElem(L,i,*e): 将线性表L中的第i个位置元素值返回给e。

LocateElem(L,e): 在线性表L中查找与给定值e相等的元素, 如果查找成功, 返回该元素在表中序号表示成功; 否则, 返回0表示失败。



线性表的抽象数据类型

ListInsert(*L,i,e): 在线性表L中第i个位置插入新元素e。

ListDelete(*L,i,*e): 删除线性表L中第i个位置元素, 并用e返回其值。

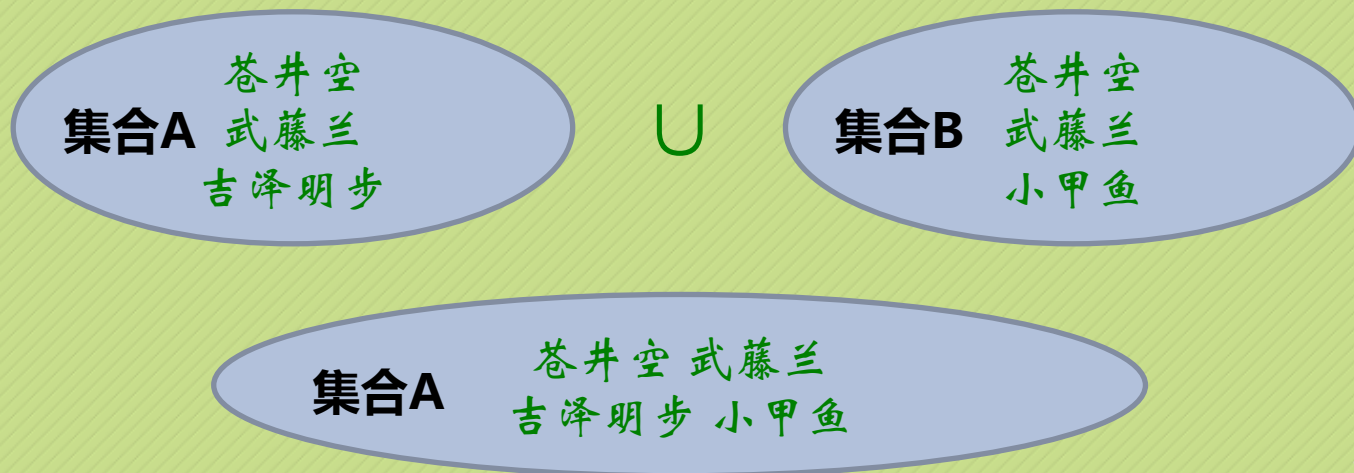
ListLength(L): 返回线性表L的元素个数。

- endADT
- 对于不同的应用, 线性表的基本操作是不同的, 上述操作是最基本的, 对于实际问题中涉及的关于线性表的更复杂操作, 完全可以用这些基本操作的组合来实现。



线性表的抽象数据类型

- 举个例子? 好吧, 比如要实现两个线性表A、B的并集操作, 即要使得集合 $A=A \cup B$ 。
- 解释下“ $A=A \cup B$ ”: 说白了, 就是把存在集合B中但不存在集合A中的元素插入到A中即可。



线性表的抽象数据类型

- 其实仔细思考一下，我们只需要循环遍历集合B中的每个元素，判断当前元素是否存在A中，若不存在，则插入A中即可。
- 综合分析，我们需要运用到几个基本的操作组合即可：
 - ListLength(L);
 - GetElem(L,i,*e);
 - LocateElem(L,e);
 - ListInsert(*L,i,e);



线性表的抽象数据类型

- 参考实现代码段: `union.c`

