

2.1 线性表及其实现

2019年5月3日, 星期五 下午 08:42

1. 以多项式为例 $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$,

① 用数组表示, 系数为数组中的值, x 的指数为数组下标.

② 用结构数组表示, 数组分量由系数 a_i 与指数 i 组成的结构.
构成非0项
排序按指数 降序排列

③ 链表

coef	expon	link
------	-------	------

```
typedef struct polyNode *Polynomial;  
struct polyNode{  
    int coef;  
    int expon;  
    Polynomial link; }
```

多项式问题的启示:

- 同一个问题有不同的表示存储方法; //存储形式不同, 数组链表balabal
- 有共性问题: 有序线性序列的组织和管理; //序列体现在按指数的降序排列

线性表: 由同类的数据元素构成有序序列的线性结构

- 元素个数即为线性表的长度;
- 线性表没有元素时, 称为空表;
- 表起始位置为表头, 结束位置为表尾

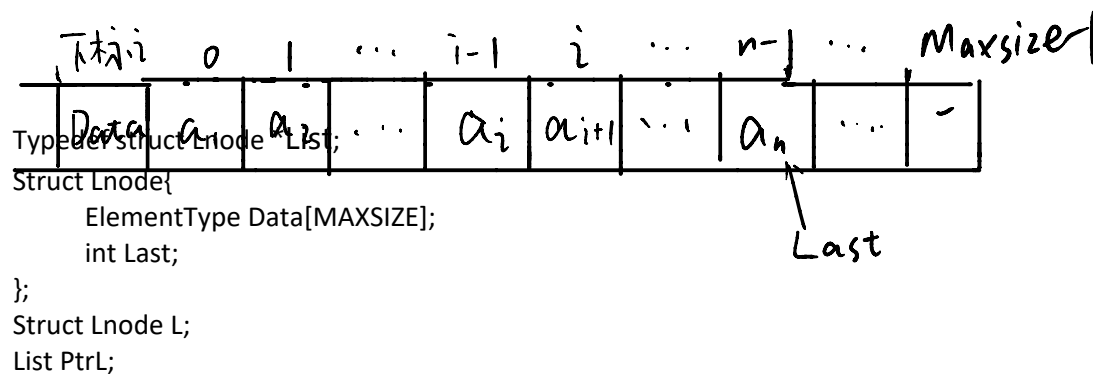
线性表的抽象数据类型描述:

类型名称: 线性表 (list)

数据对象集: 线性表是 n (≥ 0) 个元素构成的有序数列 (a_1, a_2, \dots, a_n)

操作集: 线性表 $L \in \text{list}$, 整数 i 表示位置, 元素 $x \in \text{ElementType}$, 基本操作主要有:

- List MakeEmpty()//初始化一个空线性表
- ElementType Findkth(int K, List L);//根据位序 k , 返回相应元素;
- Int Find(ElementType X, List L);//在线性表 L 中查找 X 第一次出现的位置;
- Void insert(ElementType X, int I, List L);//在线性表 L 中的第 i 个位置插入元素 X ;
- Void delete(int I, List L);//删除线性表 L 中第 i 个元素;
- Int Length(List L);//返回线性表 L 的表长 n ;



访问下标为i的元素：L.Data[i]或PtrL->Data[i];

线性表的长度：L.Last+1或PtrL->Last+1;

主要操作的实现：

- List MakeEmpty(){
List ptrL;
ptrL = (list)malloc(sizeof(struct LNode));
ptrL->last = -1;
Return ptrL;
}//初始化
- int Find(ElementType X,List PtrL){
int i=0;
While(i<=PtrL->Last&&ptr->data[i]!=x){
i++;
}
If(i>ptr->Last){
Return -1;//若没找到，返回-1
}else{
Return i;//若找到，返回i位置
}
}//时间复杂度o(n),因为比较次数(1+n)/2