



石家莊鐵道大學
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

线性表

顺序表的基本操作

—初始化、查找等

主讲：刘辉

目录

- ◆ 1 顺序表的基本操作
- ◆ 2 顺序表的初始化等操作
- ◆ 3 顺序表的查找



一、顺序表的基本操作

- ◆ 初始化线性表L `InitList(&L)`
- ◆ 销毁线性表L `DestoryList(&L)`
- ◆ 清空线性表L `ClearList(&L)`
- ◆ 求线性表L的长度 `ListLength(L)`
- ◆ 判断线性表L是否为空 `IsEmpty(L)`
- ◆ 获取线性表L中的某个数据元素内容 `GetElem(L,i,&e)`
- ◆ 检索值为e的数据元素 `LocateElem(L,e)`
- ◆ 在线性表L中插入一个数据元素 `ListInsert(&L,i,e)`
- ◆ 删除线性表L中第i个数据元素 `ListDelete(&L,i,&e)`

二、顺序表的初始化等操作

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

◆ 初始化

- 构造一个空的顺序表

```
Status InitList_Sq(SqList &L) { //构造一个空的顺序表L
```

```
L.elem=new ElemType[MAXSIZE]; //为顺序表分配空间
```

```
if(!L.elem) exit(OVERFLOW); //存储分配失败
```

```
L.length=0; //空表长度为0
```

```
return OK;
```

```
}
```

将顺序表的长度设为0

L.Elem指向顺序表的首地址

二、顺序表的初始化等操作

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

◆ 顺序表的销毁

- 释放顺序表所占用的空间

```
void DestroyList(SqList &L)
{
    //释放存储空间
    if (L.elem) delete(L.elem);
}
```

二、顺序表的初始化等操作

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

◆ 清空顺序表

- 顺序表的元素个数置为0

```
void ClearList(SqList &L)
```

```
{
```

```
    //将线性表的长度置为0
```

```
    L.length=0;
```

```
}
```

二、顺序表的初始化等操作

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

- ◆ 求顺序表的长度
返回顺序表数据元素的个数

```
int GetLength(SqList L)
{
    return (L.length);
}
```

二、顺序表的初始化等操作

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

◆ 判断顺序表L是否为空

```
int IsEmpty(SqList L)
{
    if (L.length==0) return 1;
    else return 0;
}
```


三、顺序表的查找

◆ 获取某个位置元素的内容

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

//根据指定位置，获取相应位置数据元素的内容

```
int GetElem(SqList L, int i ,ElemType &e)
```

```
{ //判断i值是否合理，若不合理，返回ERROR
  if (i<1||i>L.length) return ERROR;
  e=L.elem[i-1]; //第i-1的单元存储着第i个数据
  return OK;
}
```

三、顺序表的查找

- 根据指定数据查找，返回数据所在的位置

//在线性表L中查找值为e的数据元素

```
int LocateElem(SqList L,ElemType e)
```

```
{
```

```
    for (i=0;i< L.length;i++)
```

```
        if (L.elem[i]==e) return i+1;
```

```
    return 0;
```

```
}
```

```
#define MAXSIZE 100
typedef struct {
    ElemType *elem;
    int length
}SqList;
```

查找算法时间效率分析 ? ? ?

三、顺序表的查找

- 根据指定数据查找，返回数据所在的位置

查找成功的平均比较次数
(p_i 为各项的查找概率)

$$ASL = \sum_{i=1}^n p_i \times c_i$$

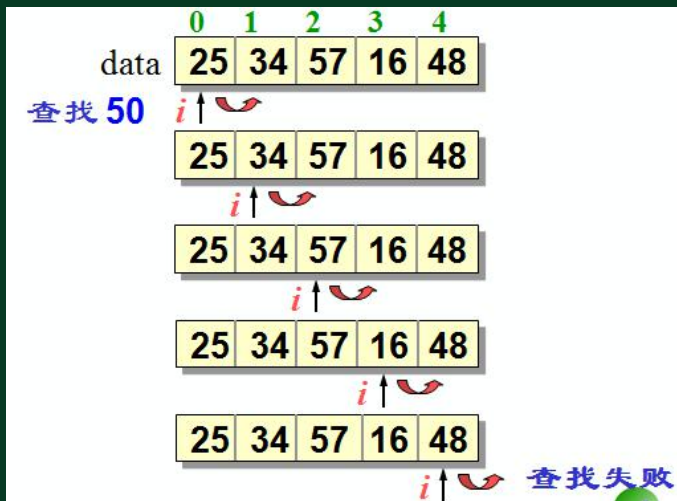
$$\begin{aligned} ASL &= \frac{1}{n} \sum_{i=1}^n i = \frac{1}{n} (1 + 2 + \dots + n) = \\ &= \frac{1}{n} * \frac{(1+n) * n}{2} = \frac{1+n}{2} \end{aligned}$$



三、顺序表的查找

- 根据指定数据查找，返回数据所在的位置

查找不成功
数据比较 n 次



小结

- 了解顺序表的初始化、销毁、清空、求长度等算法
- 掌握顺序表的查找算法
- 掌握顺序表查找算法的效率分析

谢谢！