四川師範大學
SICHUAN NORMAL UNIVERSITY

第 2 章　线 性 表

DATA STRUCTURE

计算机科学学院　刘 芳

第2章 线性表

2.1 线性表的定义

2.2 线性表的顺序表示和实现

2.3 线性表的链式表示和实现

2.4 典型示例：一元多项式的表示及相加

2.2 线性表的顺序表示和实现

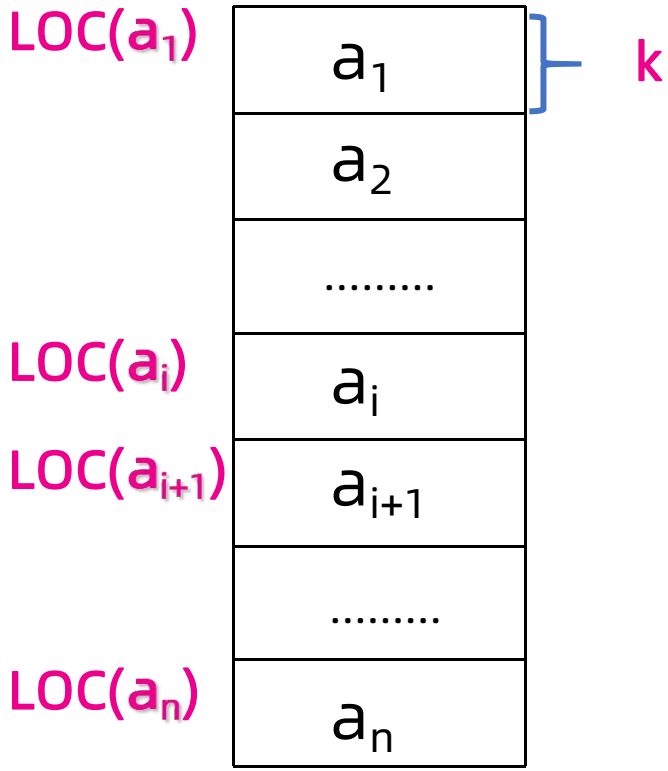
刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

定义

- 用一组地址连续的存储单元依次存放线性表中的各个元素。
  - $LOC(a_i)=LOC(a_1)+(i-1)*k$　随机存取结构
  - $LOC(a_{i+1})= LOC(a_i) + k$

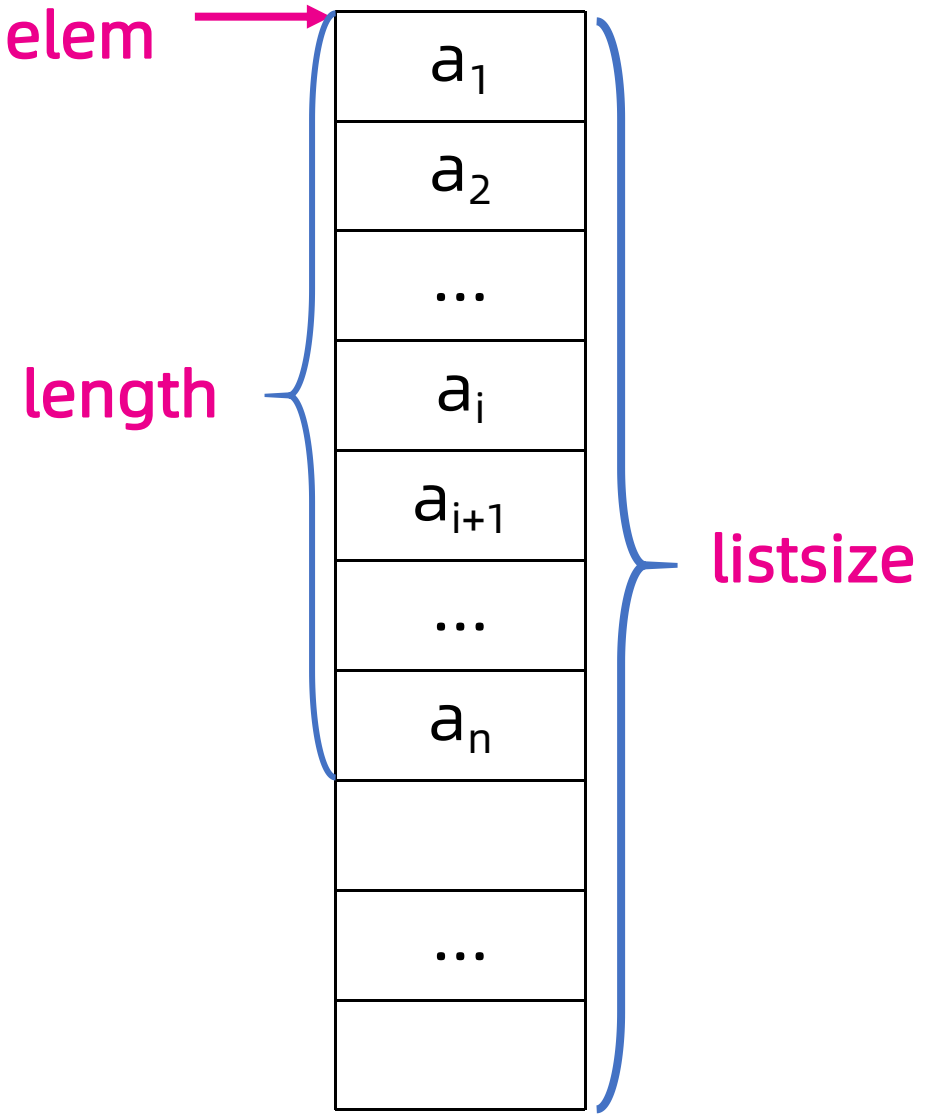
  （逻辑上相邻的元素在“物理位置上也相邻”）
- 注：线性表的顺序表示可以简称为顺序表。



顺序表的C语言描述

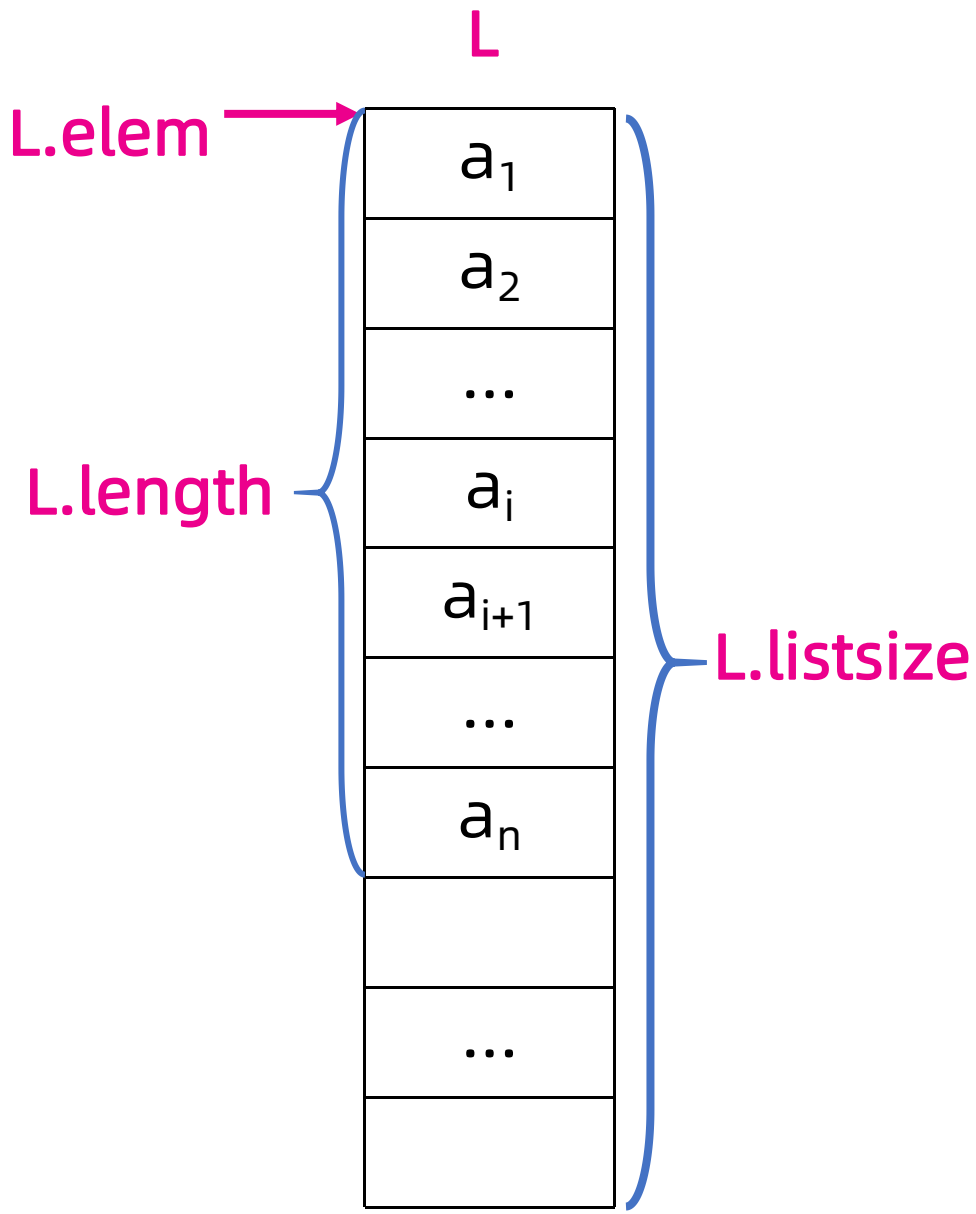
```
#define LIST_INIT_SIZE 100
#define LISTINCREMENT 10
typedef struct {
        ElemType *elem;
        int length;
        int listsize;
}SqList;

SqList L;
```



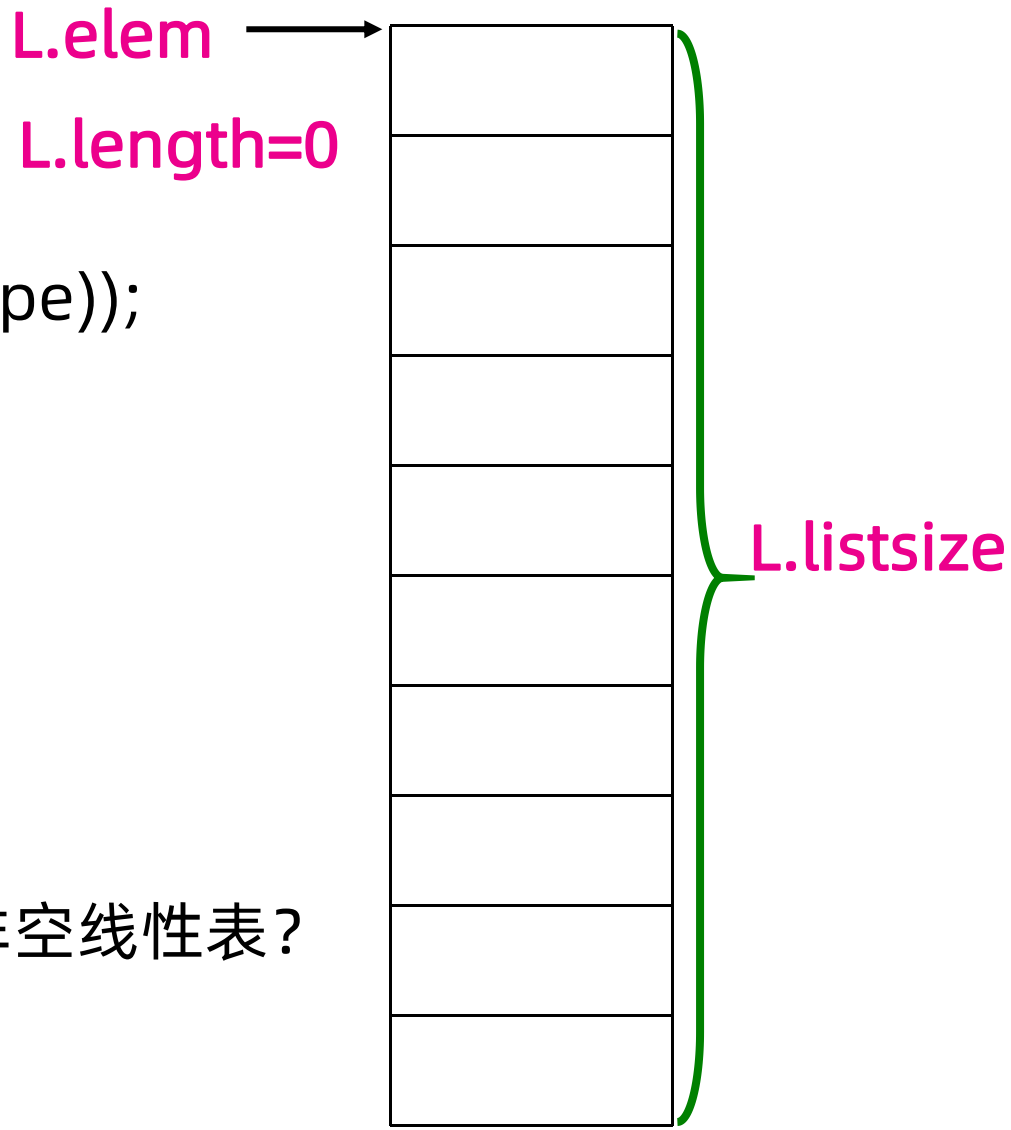
顺序表的基本操作

1. 建立 InitList(&L)
2. 输出 OutputList(L)
3. 求长度 ListLength(L)
4. 查找 LocateElem(L,e)
   GetElem(L,i,&e)
5. 插入 ListInsert(&L,i,e)
6. 删除 ListDelete(&L,i,&e)
7. 判空 ListEmpty(L)
8. 表的合并 MergeList(La,Lb,&Lc)
9. 清空表 ClearList(&L)
10. 销毁表 DestroyList(&L)



1.顺序表的建立（空表）

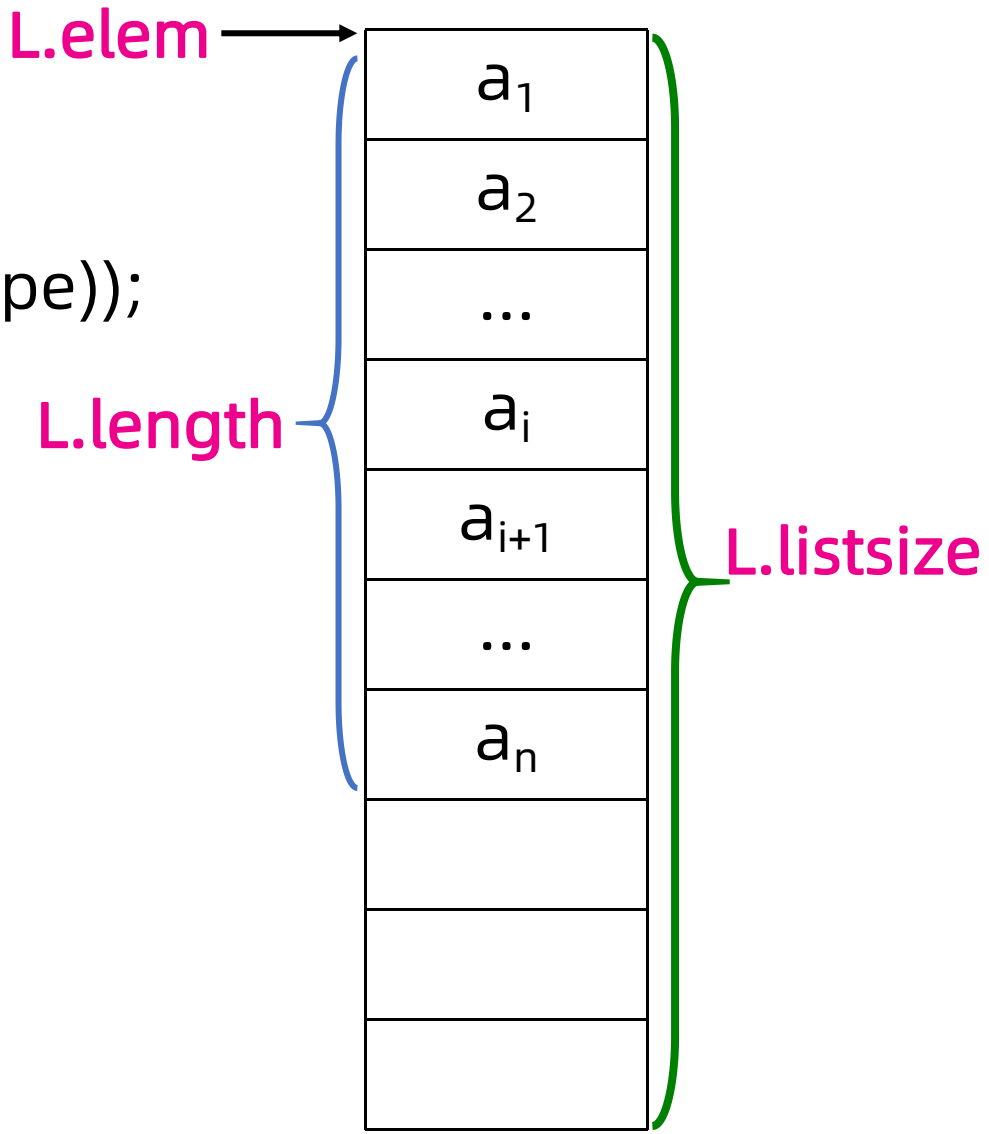
```
Status InitList(SqList & L){
    L.elem=(ElemType*)
    malloc(LIST_INIT_SIZE*sizeof(ElemType));
    if (!L.elem) exit (OVERFLOW);
    L.length=0;
    L.listsize=LIST_INIT_SIZE;
    return OK;
}
```



思考：如何建立一个具有n个数据元素的非空线性表？

1.顺序表的建立(非空表)

```
Status InitList(SqList & L){
    L.elem=(ElemType*)
    malloc(LIST_INIT_SIZE*sizeof(ElemType));
    if (!L.elem) exit (OVERFLOW);
    cin>>n;
    for (i=0;i<n;i++)    cin>>L.elem[i];
    L.length=n;
    L.listsize=LIST_INIT_SIZE;
    return OK;
}
```

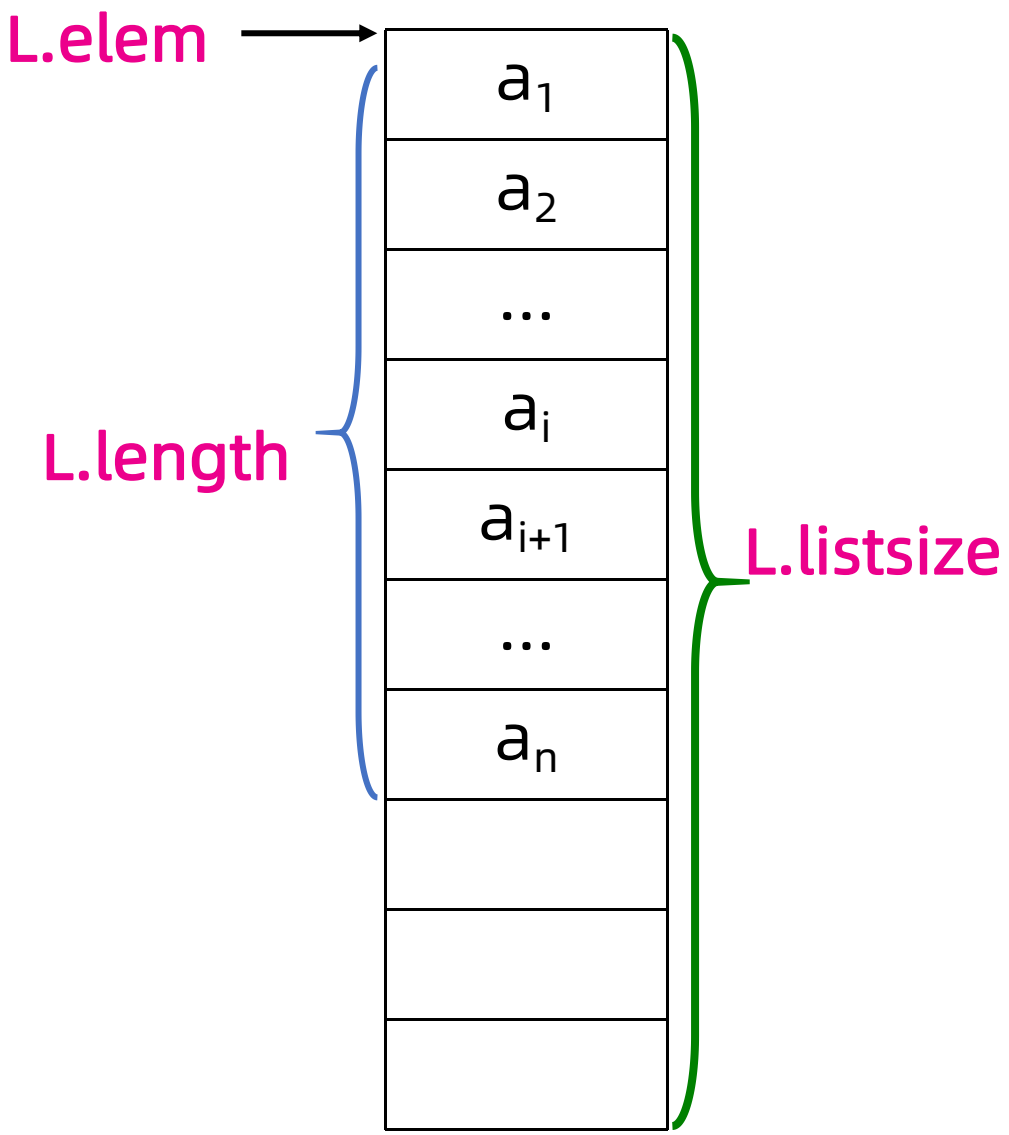


2.顺序表的输出

```
void OutPutList(SqList  L){
    for (i=0;i<L.length;i++)
        cout<<L.elem[i];
}
```

3.求表的长度

```
int ListLength(SqList  L){
    rerutn L.length;
}
```



4.顺序表的查找

按值查找

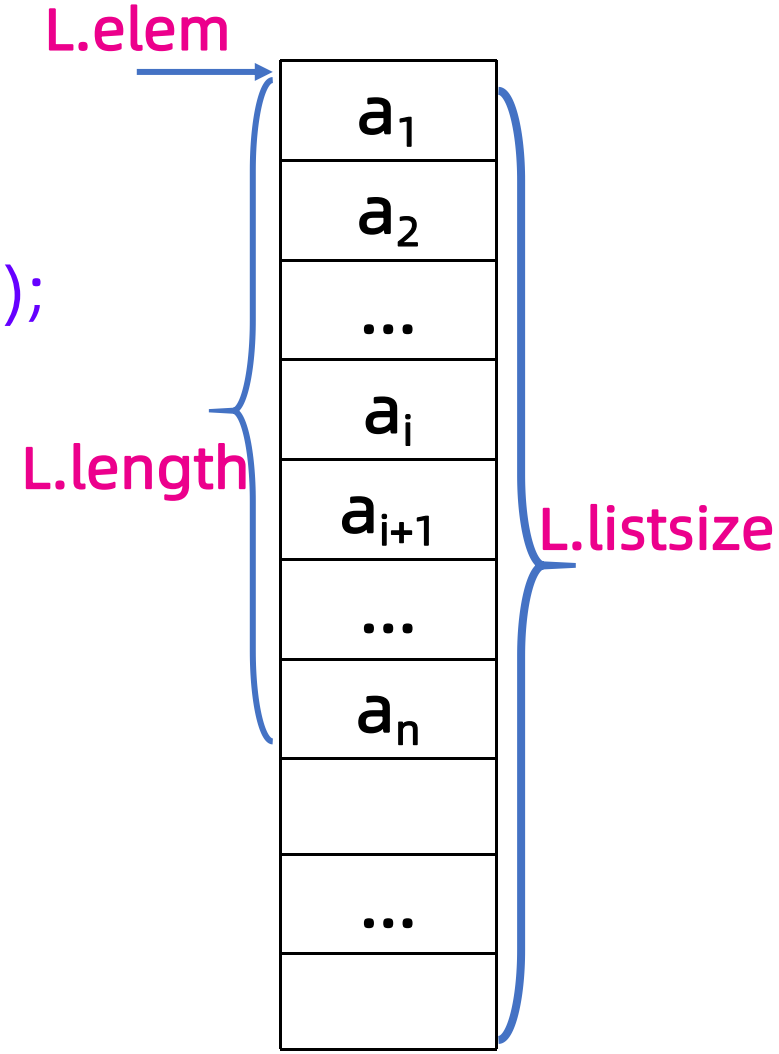
按位置查找

```
int  LocateElem(Sqlist L, ElemType  e){

  for (i=1; i<=L.length &&L.elem[i-1]!=e; i++);
  if (i<=L.length) return i;
  else return 0;

}
```

算法的时间复杂度为O(n)。



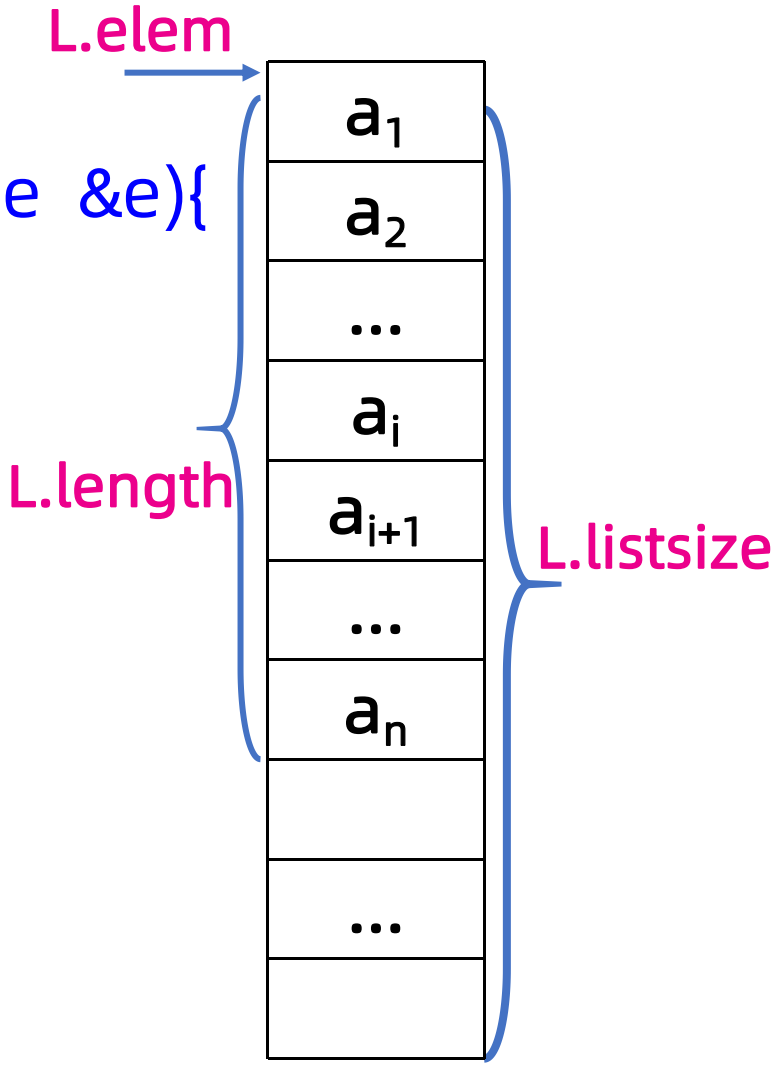
4.顺序表的查找

按值查找

按位置查找

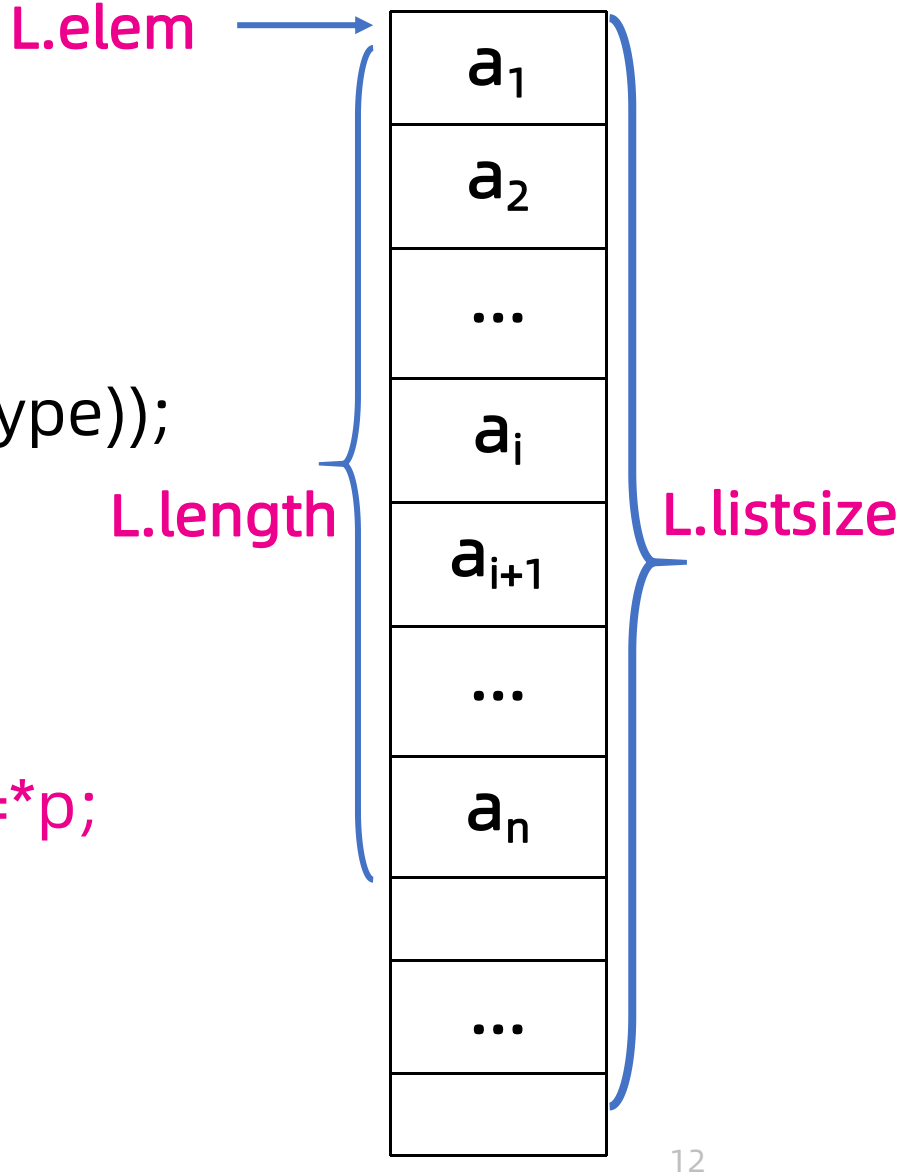
```
Status  GetElem(Sqlist L, int i,ElemType  &e){
    if (i<1||i>L.length) return ERROR;
    e=L.elem[i-1];
    return OK;
}
```

算法的时间复杂度为O(1)。



5.顺序表的插入

```
Status ListInsert(SqList &L,int i,ElemType e){
   if (i<1||i>L.length+1) return ERROR;
   if (L.length>=L.listsize){
      newbase=(ElemType *)realloc(L.elem,
              (L.listsize+LISTINCREMENT)*sizeof(ElemType));
      if(!newbase) exit (OVERFLOW);
       L.elem=newbase;L.listsize+=LISTINCREMET;
    }
    q=&(L.elem[i-1]);
    for (p=&(L.elem[L.length-1]);p>=q;- -p))  *(p+1)=*p;
    *q=e;
    ++L.length;return OK;
}
```



5.顺序表的插入操作

- 插入算法的时间复杂度分析（设L.length=n）
  - 插入算法所花的时间主要耗费在移动元素上

插入位置i	移动元素的个数c_i
1	n
2	n-1
……	……
i	n-i+1
……	……
n	1
n+1	0

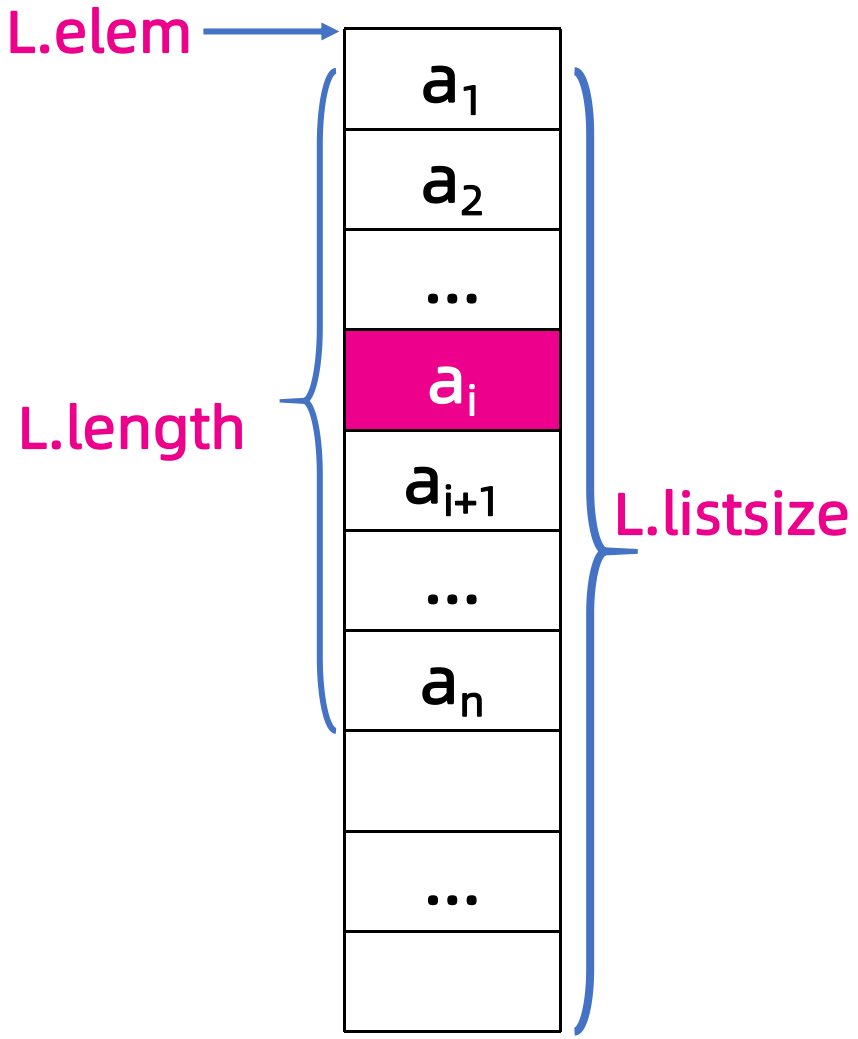
$$平均移动个数 E_{is}=\sum_{i=1}^{n+1} p_i \times c_i$$

$$=\sum_{i=1}^{n+1}[\frac{1}{n+1}\times(n-i+1)]=\frac{n}{2}$$

时间复杂度	
最好情况	O(1)
最坏情况	O(n)
平均情况	O(n)

6.顺序表的删除

```
Status ListDelete(Sqlist &L,int i,ElemType & e){
    if ((i<1)||(i>L.length)) return ERROR;
    p=&(L.elem[i-1]);
    e=*p;
    q=L.elem+L.length-1;
    for (++p; p<=q ; ++p)  *(p-1)=*p;
    --L.length;
    return OK;
}
```



6.顺序表的删除操作

- 删除算法的时间复杂度分析（设L.length=n）
  - 删除算法所花的时间主要耗费在移动元素上。

删除位置i	移动元素的个数c_i
1	n-1
2	n-2
……	……
i	n-i
……	……
n	0

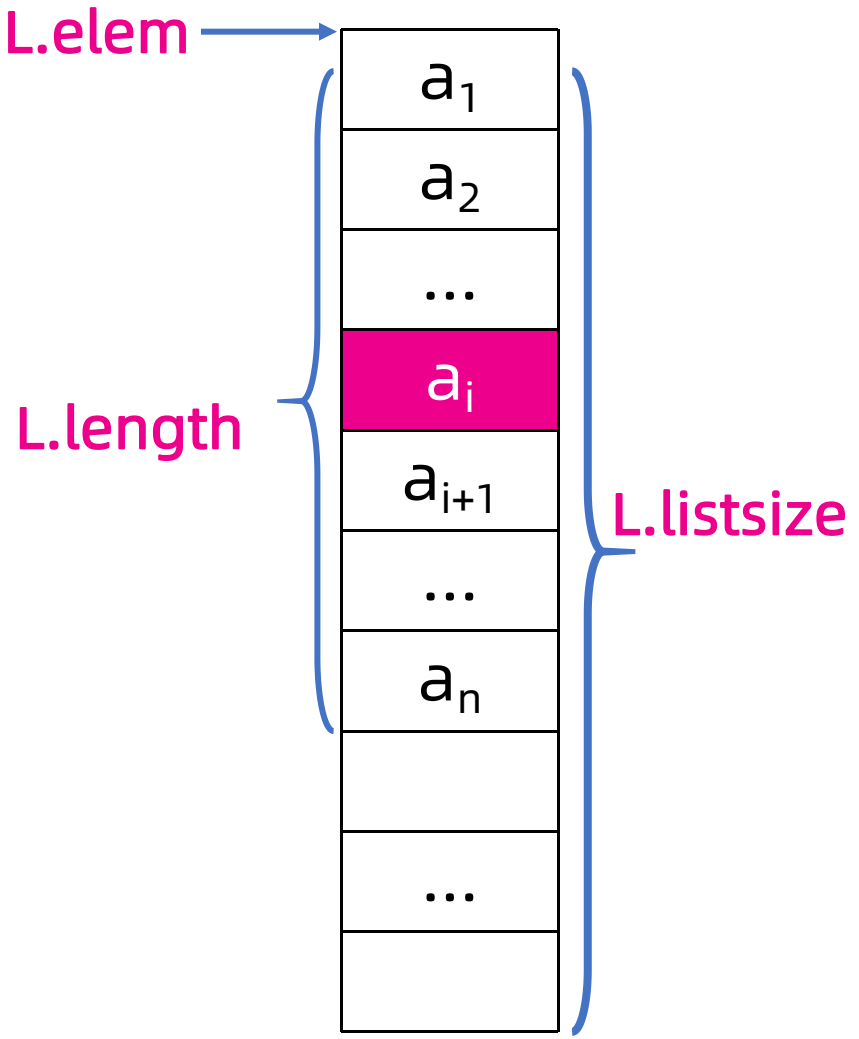
$$平均移动个数 E_{dl}=\sum_{i=1}^{n} p_i \times c_i$$

$$=\sum_{i=1}^{n}\left[\frac{1}{n}\times(n-i)\right]=\frac{n-1}{2}$$

时间复杂度	
最好情况	O(1)
最坏情况	O(n)
平均情况	O(n)

本节要点：线性表的顺序表示和实现

1. 建立 InitList(&L)
2. 输出 OutputList(L)
3. 求长度 ListLength(L)
4. 查找 LocateElem(L,e)
   GetElem(L,i,&e)
5. 插入 ListInsert(&L,i,e)
6. 删除 ListDelete(&L,i,&e)
7. 判空 ListEmpty(L)
8. 表的合并 MergeList(La,Lb,&Lc)
9. 清空表 ClearList(&L)
10. 销毁表 DestroyList(&L)



四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.