



本节主题:

线性表上的顺序查找

被查找的顺序表类型

```
#define MAXL 100
typedef int KeyType;
typedef char InfoType[10];
typedef struct
{
    KeyType key;
    InfoType data;
} NodeType;
typedef NodeType SeqList[MAXL];
```

	关键字	其他数据			
	key	data			
[0]					
[1]					
[2]					
[3]					
[4]					
[5]					
[6]					
[...]					

顺序查找思路

例

查找表：3 9 1 5 8 10 6 7 2 4

要查找的值：k=10

从表的一端开始，顺序扫描线性表，依次将扫描到的关键字和给定值k相比较

若当前扫描到的关键字与k相等，则查找成功；

若扫描结束后，仍未找到关键字等于k的记录，则查找失败。

第1次比较： 3 9 1 5 8 10 6 7 2 4

i=0

第2次比较： 3 9 1 5 8 10 6 7 2 4

i=1

第3次比较： 3 9 1 5 8 10 6 7 2 4

i=2

第4次比较： 3 9 1 5 8 10 6 7 2 4

i=3

第5次比较： 3 9 1 5 8 10 6 7 2 4

i=4

第6次比较： 3 9 1 5 8 10 6 7 2 4

i=5

查找成功，返回逻辑序号 i+1=6

算法实现及性能评价

```
typedef struct
{
    KeyType key;
    InfoType data;
} NodeType;
typedef NodeType SeqList[MAXL];
int SeqSearch(SeqList R,int n,KeyType k)
{
    int i=0;
    while (i<n && R[i].key!=k)
        i++;
    if (i>=n)
        return 0;
    else
        return i+1;
}
```

key	3	9	1	5	8	10	6	7	2	4
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

☐ 成功时的平均查找长度

$$\begin{aligned}ASL_{sq} &= \sum_{i=1}^n p_i c_i \\&= \frac{1}{n} \sum_{i=1}^n i \\&= \frac{1}{n} \times \frac{n(n+1)}{2} \\&= \frac{n+1}{2} \\&= O(n)\end{aligned}$$