

# 查找

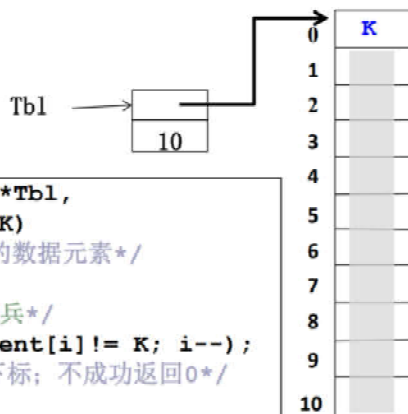
2019年6月30日 22:11

## 静态查找

### 1. 顺序查找

#### 静态查找

#### 方法1: 顺序查找



### 2.

```
int SequentialSearch (StaticTable *Tbl,
                     ElementType K)
{ /*在表Tbl[1]~Tbl[n]中查找关键字为K的数据元素*/
    int i;
    Tbl->Element[0] = K; /*建立哨兵*/
    for(i = Tbl->Length; Tbl->Element[i] != K; i--);
    return i; /*查找成功返回所在单元下标; 不成功返回0*/
}
```

顺序查找算法的时间复杂度为 $O(n)$ 。

### 3. 二分查找

假设n个数据元素的关键字满足有序（比如：小到大）并且是连续存放（数组），那么可以进行二分查找。

例1:

**[例]** 假设有13个数据元素，按关键字由小到大顺序存放。  
二分查找关键字为444的数据元素过程如下：

5	16	39	45	51	98	100	202	226	321	368	444	501
1	2	3	4	5	6	7	8	9	10	11	12	13
↑						↑					↑	
left						mid					right	

1、left = 1, right = 13; mid = (1+13)/2 = 7: 100 < 444;

2、left = mid+1=8, right = 13; mid = (8+13)/2 = 10: 321 < 444;

3、left = mid+1=11, right = 13; mid = (11+13)/2 = 12: 查找结束;

例2:

**[例]** 仍然以上面13个数据元素构成的有序线性表为例  
二分查找关键字为43的数据元素如下：

5	16	39	45	51	98	100	202	226	321	368	444	501
1	2	3	4	5	6	7	8	9	10	11	12	13
↑						↑					↑	
left						mid					right	

1、left = 1, right = 13; mid = (1+13)/2 = 7: 100 > 43;

**[例]** 仍然以上面13个数据元素构成的有序线性表为例  
二分查找关键字为 **43** 的数据元素如下：

5	16	39	45	51	98	100	202	226	321	368	444	501
1	2	3	4	5	6	7	8	9	10	11	12	13
↑ left						↑ mid						↑ right

- 1、left = 1, right = 13; mid = (1+13)/2 = 7:      **100 > 43;**
- 2、left = 1, right = mid-1= 6; mid = (1+6)/2 = 3:    **39 < 43;**
- 3、left = mid+1=4, right = 6; mid = (4+6)/2 = 5:    **51 > 43;**
- 4、left = 4, right = mid-1= 4; mid = (4+4)/2 = 4:    **45 > 43;**
- 5、left = 4, right = mid-1= 3; **left > right ?** 查找失败，结束;

二分查找算法具有对数的时间复杂度 $O(\log N)$

```
int BinarySearch(StaticTable* Tbl, ElementType K)
```

```
{ /*在表Tbl中查找关键字为K的数据元素*/
```

```
    int left, right, mid, NotFound = -1;
```

```
    left = 1; /*初始左边界*/
```

```
    right = Tbl->Length; /*初始右边界*/
```

```
    while (left <= right)
```

```
    {
```

```
        mid = (left + right) / 2; /*计算中间元素坐标*/
```

```
        if (K < Tbl->Element[mid]) right = mid - 1; /*调整右边界*/
```

```
        else if (K > Tbl->Element[mid]) left = mid + 1; /*调整左边界*/
```

```
        else return mid; /*查找成功，返回数据元素的下标*/
```

```
    }
```

```
    return NotFound; /*查找不成功，返回-1*/
```

```
}
```

动态查找