

索引|表查找 | 索引表的基本概念

3. 索引表的顺序查找算法思想描述

首先根据待查找关键字在索引表当中定位块。定位的方法是：只要 $key >$ 索引块 i 的最大关键值，则 $i++$ ，定位下一个索引项；直到定位到索引块，或者把索引项都定位完也没有比 key 关键字大的索引项。

如果定位到块，则在块内部进行顺序查找。

3. 索引表的顺序查找算法

```
int IndexSequelSearch(IndexType ls[], DataType s[], int m, KeyType key){  
    /*索引表为ls[0]-ls[m-1], 顺序表为s */  
    i=0;  
    while(i<m && key>ls[i].key) i++; /*块间查找*/  
    if(i==m) return -1; /*查找失败*/  
    else{ /*在块内顺序查找*/  
        j=ls[i].Link;  
        while(Key!=s[j].key && j<ls[i+1].Link) j++;  
        if(key == s[j].key) return j; /*查找成功*/  
        else return -1; /*查找失败*/  
    }  
}
```

3. 索引表的顺序查找算法

```
int IndexSequelSearch(IndexType ls[], DataType s[], int m, KeyType key){  
    /*索引表为ls[0]-ls[m-1], 顺序表为s */  
    i=0;  
    while(i<m && key>ls[i].key) i++; /*块间查找*/  
    if(i==m)return -1; /*查找失败*/  
    else{ /*在块内顺序查找*/  
        j=ls[i].Link;  
        while(Key!=s[j].key && j<ls[i+1].Link) j++;  
        if(key == s[j].key)return j; /*查找成功*/  
        else return -1; /*查找失败*/  
    }  
}
```

3. 索引表的顺序查找算法

```
int IndexSequelSearch(IndexType ls[], DataType s[], int m, KeyType key){
```

```
/*索引表为ls[0]-ls[m-1], 顺序表为s*/
```

```
    i=0;
```

```
    while(i<m && key>ls[i].key) i++; /*块间查找*/
```

```
    if(i==m) return -1; /*查找失败*/
```

```
    else{ /*在块内顺序查找*/
```

```
        j=ls[i].Link;
```

```
        while(Key!=s[j].key && j<ls[i+1].Link) j++;
```

```
        if(key == s[j].key) return j; /*查找成功*/
```

```
        else return -1; /*查找失败*/
```

```
    }
```

```
}
```

```
typedef struct IndexType  
{  
    KeyType key;  
    int Link;  
} IndexType;
```

- 索引顺序查找的ASL?
- $ASL = ASL(\text{索引表}) + ASL(\text{块内})$

4. 索引表的顺序查找性能分析

$$ASL_{bs} = L_b + L_w = \frac{1}{b} \sum_{j=1}^b j + \frac{1}{s} \sum_{i=1}^s i = \frac{1}{2} \left(\frac{n}{s} + s \right) + 1$$

其中， n 为表长，均匀分为 b 块，每块含有 s 个记录