



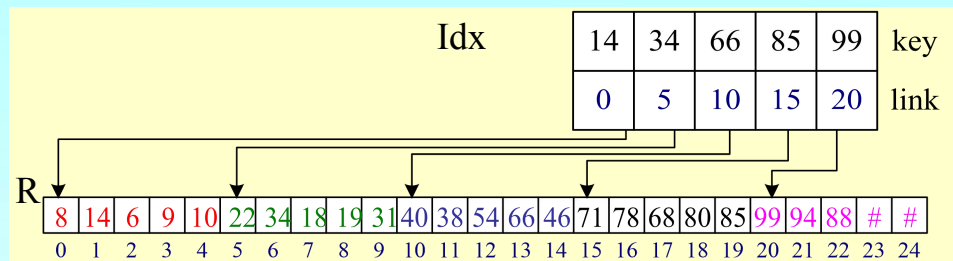
本节主题:

分块查找

分块查找

- 别名：索引顺序查找，一种介于顺序查找和二分查找之间的查找方法。
- 例如：设有一个线性表，其中包含23个记录
 - 关键字序列：8,14,6,9,10,22,34,18,19,31,40,38,54,66,46,71,78,68,80,85,99,94,88
 - 方案：将23个记录分为5块，每块中有5个记录.....
- 思路

- 将数据表 $R[0..n-1]$ 均分为 b 块，前 $b-1$ 块中记录个数为 $s=\lceil n/b \rceil$ ，最后一块即第 b 块的记录数小于等于 s ；
- 每一块中的关键字不一定有序，但前一块中的最大关键字必须小于后一块中的最小关键字，即要求表是“分块有序”的；
- 抽取各块中的最大关键字，构建索引表 $Idx[0..b-1]$
 - $Idx[i] (0 \leq i \leq b-1)$ 中附加存放着第 i 块在表 R 中的起始位置
 - 由于表 R 是分块有序的，所以索引表是一个递增有序表。



存储结构

```
#define MAXL 100    //数据表的最大长度
#define MAXI 20     //索引表的最大长度
```

```
typedef int KeyType;
typedef char InfoType[10];
typedef struct
```

```
{
    KeyType key; //关键字
    InfoType data; //其他数据
```

```
} NodeType;
```

```
typedef NodeType SeqList[MAXL]; //顺序表类型
```

```
SeqList R;
```

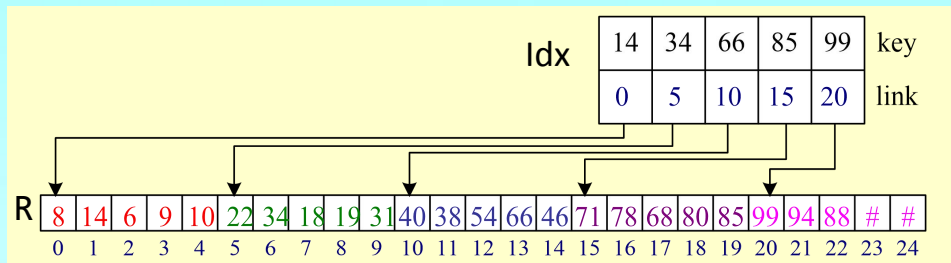
```
typedef struct
```

```
{
    KeyType key;           //KeyType为关键字的类型
    int link;              //对应块的起始下标
```

```
} IdxType;
```

```
typedef IdxType IDX[MAXI]; //索引表类型
```

```
IDX idx;
```



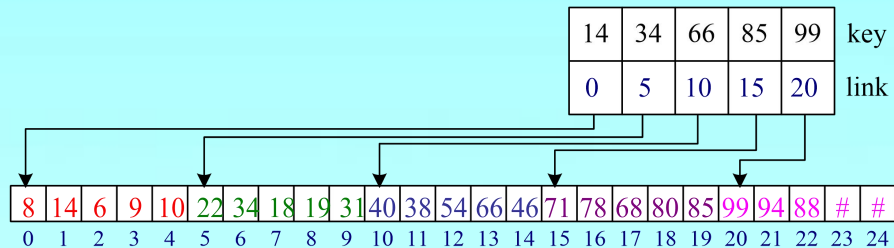
分块查找系统

1. 建立和维护索引表和数据表;
2. 利用索引表和数据表查找数据。

查找算法实现及复杂度分析

```
int IdxSearch(IDX I,int m,SeqList R,int n,KeyType k)
```

```
{
    int low=0,high=m-1,mid,i;
    int b=n/m;
    //在索引表中“折半”查找
    while (low<=high)
    {
        mid=(low+high)/2;
        if (I[mid].key>=k)
            high=mid-1;
        else
            low=mid+1;    //找到位置是high+1
    }
    //到数据表中存储查找
    i=I[high+1].link;
    while (i<=I[high+1].link+b-1 && R[i].key!=k) i++;
    if (i<=I[high+1].link+b-1)
        return i+1;
    else
        return 0;
}
```



空间性能：需要额外存储空间(索引表)

时间性能：分块查找的平均查找长度

折半查找索引表时

$$\begin{aligned} ASL_{blk} &= ASL_{bn} + ASL_{sq} \\ &= \log_2(b+1) - 1 + \frac{s+1}{2} \\ &\approx \log_2(n/s+1) + \frac{s}{2} \end{aligned}$$

顺序查找确定块时

$$\begin{aligned} ASL'_{blk} &= ASL_{bn} + ASL_{sq} \\ &= \frac{b+1}{2} + \frac{s+1}{2} \\ &= \frac{s^2 + 2s + n}{2s} \end{aligned}$$

每块的长度 s 越小, ASL_{blk} 值越小。

当 $s = \sqrt{n}$ 时, $ASL'_{blk} = \sqrt{n} + 1$ 取极小值。

例 应用分块查找

□ 问题：对于具有10000个元素的文件

(1)若采用分块查找法查找，并用顺序查找来确定元素所在的块，则

- ④ 分成几块最好？
- ④ 每块的最佳长度为多少？
- ④ 此时的平均查找长度为多少？
- ④ 相同参数，折半查找确定块呢？

(2)采用分块查找法，假定每块长度为 $s=20$ ，求平均查找长度。

(3)若直接采用顺序查找和折半查找，其平均查找长度各是多少？

折半查找确定块时

$$\begin{aligned}ASL_{blk} &= ASL_{bn} + ASL_{sq} \\&= \log_2(b+1) - 1 + \frac{s+1}{2} \\&\approx \log_2(n/s + 1) + \frac{s}{2}\end{aligned}$$

顺序查找确定块时

$$\begin{aligned}ASL'_{blk} &= ASL_{bn} + ASL_{sq} \\&= \frac{b+1}{2} + \frac{s+1}{2} \\&= \frac{s^2 + 2s + n}{2s}\end{aligned}$$

- (1) 每块中最佳元素个数 $s = \sqrt{10000} = 100$ ，
总的块数 $b = \lceil n/s \rceil = 100$ ，
顺序查找索引表， $ASL = (b+s)/2 + 1 = 100 + 1 = 101$
折半查找索引表， $ASL = \log_2(b+1) + \frac{s}{2} = \log_2 101 + 50 = 57$

- (2) 顺序查找索引表， $ASL = (b+s)/2 + 1 = 260 + 1 = 261$
折半查找索引表， $ASL = \log_2(b+1) + \frac{s}{2} = \log_2 501 + 10 = 19$

- (3) 直接用顺序查找， $ASL = (10000+1)/2 = 5000.5$
直接用折半查找， $ASL = \log_2 10001 - 1 = 13$

查找的问题，不仅仅是查找

❏ 基于顺序表的**静态查找**

❏ 顺序查找

❏ 折半查找

❏ 利用索引的查找

❏ 分块查找

❏ 基于树表的**动态查找**

❏ 高效率的查找

❏ 数据更新频繁时，保证数据更新及查找的综合效率

❏ 对只能存储在文件中的大规模数据的查找

平衡二叉树

二叉排序树

B+树

B-树

