

9.1.4 索引顺序表的查找

1. 基本索引查找

- 一般地，索引存储结构需要在数据表基础上建立一个关于索引项的索引表。
- 索引表的结构为：（索引关键字，该关键字记录在数据表中的相对地址），其中索引关键字项有序排列。

索引表

地址	索引关键字	对应地址
0	k_1	1
1	k_2	0
⋮	⋮	⋮
$n-1$	k_{n-1}	...

数据表

地址	索引关键字	其他数据项
0	k_2	...
1	k_1	...
⋮	⋮	⋮
$n-1$	k_{n-1}	...

索引存储结构 = 数据表 + 索引表

索引表

key	addr
03	2k
08	1k
17	6k
24	4k
47	5k
51	7k
83	0
95	3k

数据表

职工号	姓名	性别	职务	婚否	
83	林达	女	教师	已婚	...
08	陈洱	男	教师	已婚	...
03	张珊	男	教务员	已婚	...
95	李斯	女	实验员	未婚	...
24	何武	男	教师	已婚	...
47	王璐	男	教师	已婚	...
17	刘淇	男	实验员	未婚	...
51	岳跋	女	教师	未婚	...

索引表

地址	索引关键字	对应地址
0	k_1	1
1	k_2	0
⋮	⋮	⋮
$n-1$	k_{n-1}	...

数据表

地址	索引关键字	其他数据项
0	k_2	...
1	k_1	...
⋮	⋮	⋮
$n-1$	k_{n-1}	...

在索引存储结构中查找关键字为 k 的记录的过程：

- 先在索引表中查找，由于索引表是按关键字有序排列的，所以可以采用折半查找。
- 当找到后通过其对应地址直接在数据表中找到其记录。

2. 分块查找

- 若数据表中的数据呈现这样的规律：数据表可以分成若干块，每一块中的元素是无序的，但块与块之间元素是有序的，即前一块中的最大关键字小于（或大于）后一块中的最小（或最大）关键字值。

DataSet = { 8, 14, 6, 9, 10, 22, 34, 18, 19, 31, 40, 38, 54, 66, 46, 71, 78, 68, 80, 85, 100, 94, 88, 96, 87 }

DataSet = { 8, 14, 6, 9, 10, 22, 34, 18, 19, 31, 40, 38, 54, 66, 46, 71, 78, 68, 80, 85, 100, 94, 88, 96, 87 }

2. 分块查找

用索引表表示这种特性

- 索引表中的一项对应数据表中的一块，索引项由关键字域和链域组成，关键字域存放相应块的**最大关键字**，链域存放指向本块**第一个元素的指针**，索引表按关键字值递增（或递减）顺序排列。
- 在这种索引结构中的查找称为**分块查找**。

索引表

...				...	key
...				...	link

分块查找过程分为两步进行：

- 首先确定待查找的元素属于哪一块，即查找其所在的块。
- 然后在块内查找相应的元素。
- 由于索引表是递增有序的，可以对索引表进行折半查找，当索引表中元素个数（即分块的块数）较少时，也可以对索引表采用顺序查找方法。在进行块内查找时，由于块内元素无序，所以只能采用顺序查找方法。

【示例-1】有一个关键字序列为：

(9, 22, 12, 14, 35, 42, 44, 38, 48, 60, 58, 47, 78, 80, 77, 82) ,
给出分块查找的索引结构和查找算法。

解:

(9, 22, 12, 14, 35, 42, 44, 38, 48, 60, 58, 47, 78, 80, 77, 82)

数据表

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	22	12	14	35	42	44	38	48	60	58	47	78	80	77	82

□ 整个数据是无序的

□ 可以划分为4块:

● 9, 22, 12, 14

● 35, 42, 44, 38

● 48, 60, 58, 47

● 78, 80, 77, 82

块之间是有序的

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	22	12	14	35	42	44	38	48	60	58	47	78	80	77	82

索引表

22	0
44	4
60	8
82	12

主数据表

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	22	12	14	35	42	44	38	48	60	58	47	78	80	77	82

索引表

22	0
44	4
60	8
82	12

主数据表

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	22	12	14	35	42	44	38	48	60	58	47	78	80	77	82

查找58:

- ① 在索引表中找到关键字恰好大于等于58的块: 第3块
- ② 在第3块中顺序查找: 逻辑序号为11。

分块查找实际上进行两次查找!

性能分析

- 索引顺序查找的查找成功时的平均查找长度

$$ASL_{bs} = ASL_b + ASL_w$$

- ASL_b 是在索引表中查找子表位置的平均查找长度;
- ASL_w 是在子表内查找元素位置的查找成功的平均查找长度。

- 设把长度为 n 的表分成均等的 b 个子表, 每个子表 s 个元素, 则 $b = \lceil n/s \rceil$ 。又设表中每个元素的查找概率相等, 则每个子表的查找概率为 $1/b$, 子表内各元素的查找概率为 $1/s$ 。
- 若对索引表和子表都用顺序查找, 则索引顺序查找的查找成功时的平均查找长度为
 - $$\begin{aligned} ASL_{bs} &= (b+1)/2 + (s+1)/2 = (b+s)/2 + 1 \\ &= (n/s + s)/2 + 1 \end{aligned}$$
- 索引顺序查找的平均查找长度与表中的元素个数 n 有关, 与每个子表中的元素个数 s 有关。在给定 n 的情况下, s 应选择多大?

- 用数学方法可导出：

当 $s = \sqrt{n}$ 时， ASL_{bs} 取极小值 $\sqrt{n} + 1$ 。

这个值比顺序查找强，但比折半查找差。

- 若采用折半查找确定元素所在的子表，则查找成功时的平均查找长度为

- $ASL_{bs} = ASL_b + ASL_w$

- $\approx \log_2 (b+1) - 1 + (s+1)/2$

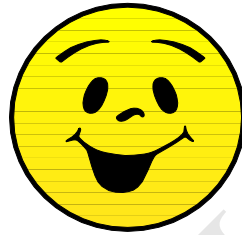
- $\approx \log_2 (1 + n/s) + (s-1)/2$

【示例-2】 设数据序列中有100个元素，待查找元素的关键字 $k=47$ 。如果在查找过程中，和 k 进行比较的元素依次是27，47，16，52，47，则所采用的查找方法可能是（ ）。

- A. 顺序查找 B. 折半查找 C. 分块查找 D. 都不是

解：

- 只有分块查找需要进行两次查找。
- 第一次与47的比较是在索引表中查找，第2次与47的比较是在对应块中查找。
- 答案为C



— END —