



第8讲 数据库索引(2)



本讲学习什么？

基本内容

1. 为什么需要索引与什么是索引
2. 索引的简单分类
3. B+树索引
4. 散列索引

重点与难点

- 理解索引的作用，掌握应用索引改进数据库查询性能的方法
- 理解不同类型索引的概念：稠密索引与稀疏索引，主索引与辅助索引，聚簇索引与非聚簇索引，倒排索引，多级索引等
- 理解B+树索引，怎样建立、维护和利用B+树索引(算法层面)
- 理解散列索引，包括静态散列索引与动态散列索引(算法层面)

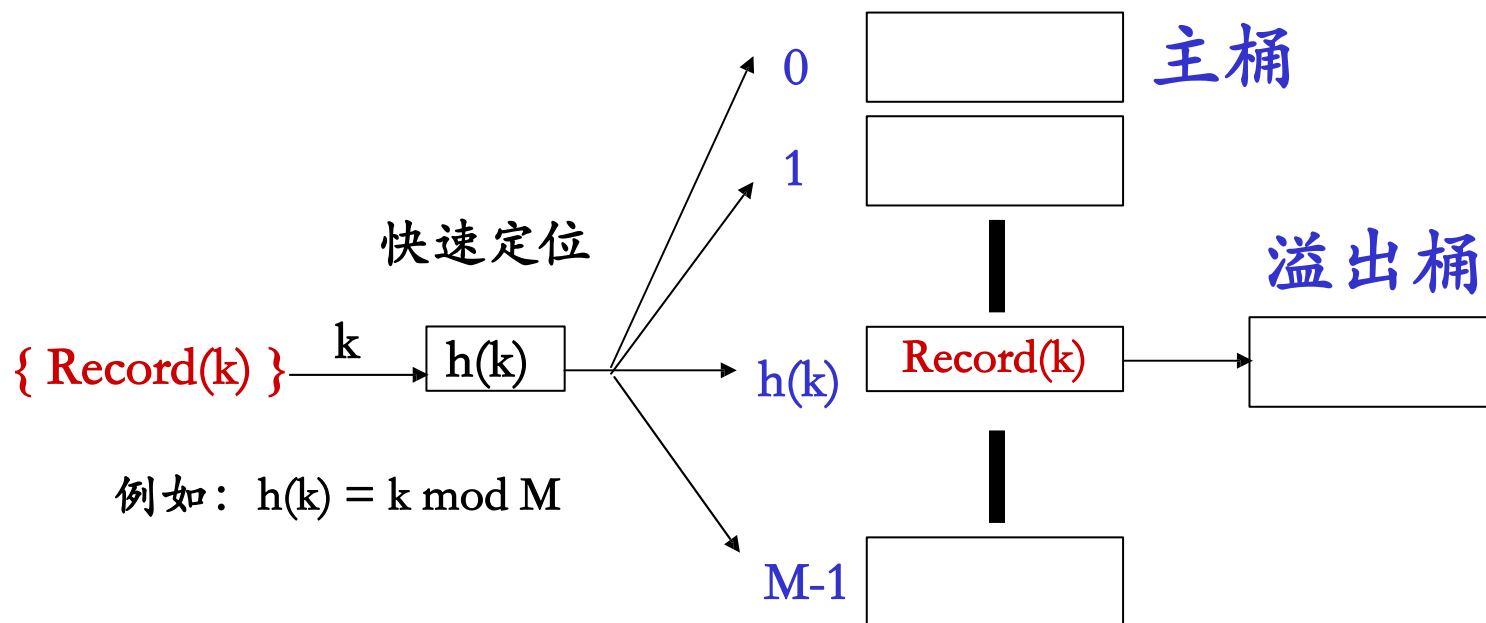


散列索引 (1)散列的基本概念

➤ 散列

- ✓ 有M个桶，每个桶是有相同容量的存储(可以是内存页，也可以是磁盘块)
- ✓ 散列函数 $h(k)$ ，可以将键值k映射到 $\{0, 1, \dots, M-1\}$ 中的某一个值
- ✓ 将具有键值k的记录Record(k) 存储在对应h(k)编号的桶中

➤ 目标：选择一个合适的散列函数，将一个Record集合(每个Record都包含一个关键字k)均匀地映射到M个桶中。即：对于集合中任一个关键字，经散列函数映射到地址集合中任何一个地址的概率是近乎相等的。





散列索引 (2)散列索引

➤ 散列索引

- ✓ 内存数据可采用散列确定存储页，主文件可采用散列确定存储块，索引亦可采用散列确定索引项的存储块
- ✓ M个桶。一个桶可以是一个存储块，亦可是若干个连续的存储块。

示例：假设 1 存储块可存放 2 个键值及其指针

$M=4$; 1 个桶为 1 个存储块

$h(x)$ 满足：

$h(e)=0$; $h(b)=h(f)=h(s)=1$

$h(g)=2$; $h(a)=h(c)=3$

问：如何查询键值 a 的索引项？

计算 $h(a)=3$

读取 3 号桶，获得键值 a 的索引项。

需要 1 个磁盘块读取

0	e
1	b f
2	g
3	a c



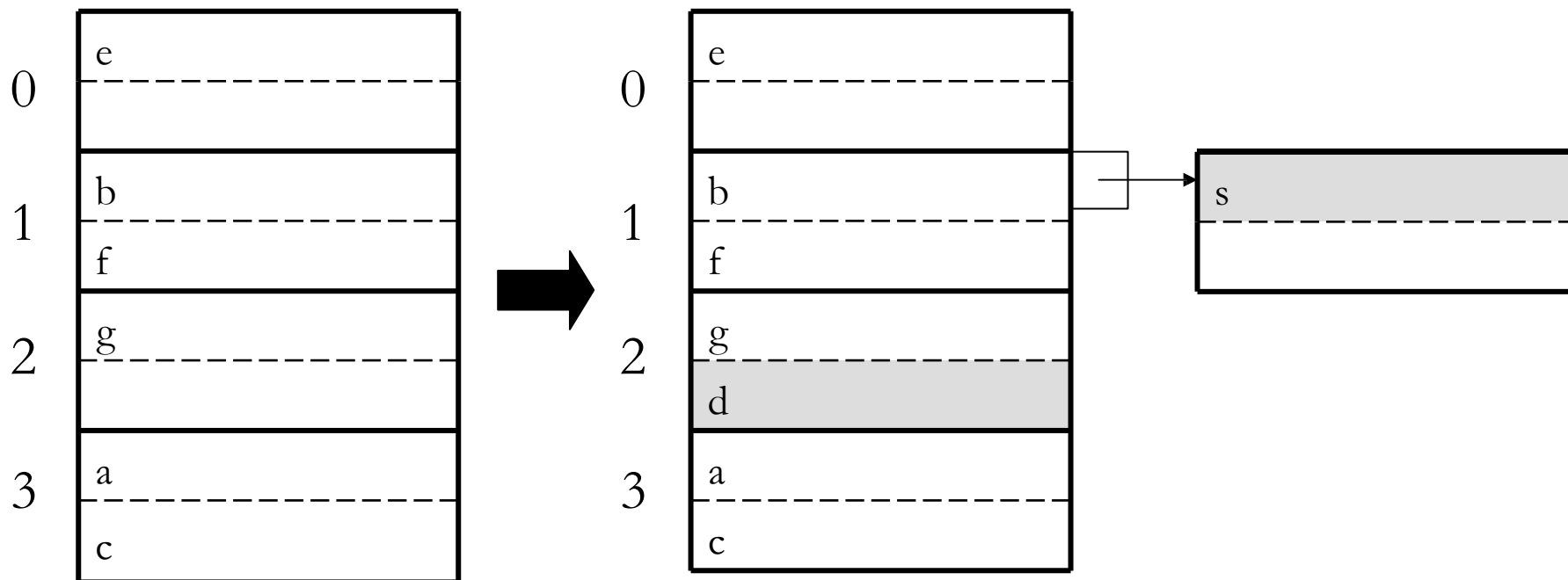
散列索引 (3)散列索引的插入和删除

问：如何插入键值d的索引项？

- ✓ 计算 $h(d)=2$
- ✓ 如2号桶有空间，则将索引项d插入2号桶中

问：如何插入键值s的索引项？

- ✓ 计算 $h(s)=1$
- ✓ 1号桶无空间，则申请一溢出桶，插入s

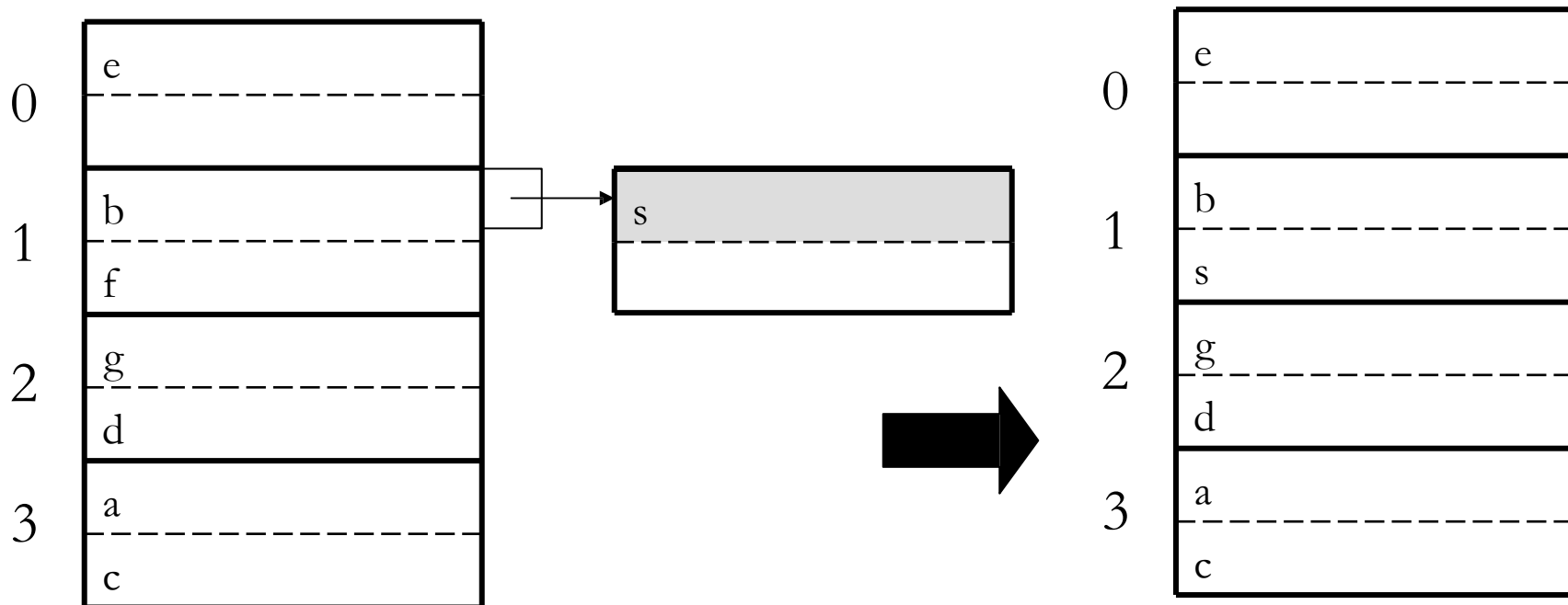




散列索引 (3)散列索引的插入和删除

问：如何删除键值f.

- ✓ 计算 $h(f)=1$
- ✓ 删除1号桶中的键值f
- ✓ 将溢出桶中的s合并到主桶中，删除溢出桶





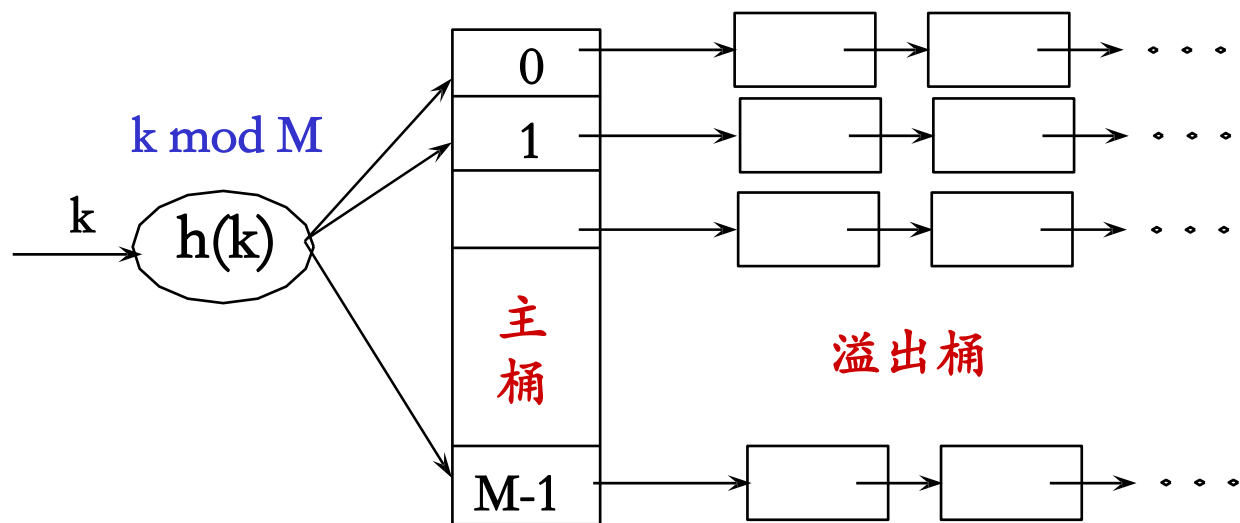
散列索引 (4)散列的问题

➤ 散列索引的目标:

最好是没有溢出桶，每一个散列值仅有一个桶。读写每一个键值都只读写一个存储块。

➤ 均匀分布如何做到?

期望将所有数据分布均匀地存储于M个桶中，使每一个桶的数据成为具有某种特征值 $h(k)$ 的数据集合。
→散列函数的选择。



➤ 桶的数目M如何确定?

在键值几倍于桶的数目时，每个散列值都可能多于一个桶，形成一个主桶和多个溢出桶的列表，此时需要二次检索：先散列找到主桶号，再依据链表逐一找到每个溢出桶。---桶的数目的确定。



散列索引 (4)散列的问题

➤ 静态散列索引

- ✓ 桶的数目 M 是固定值
- ✓ 如果桶的数目 M 不变:
 - M 过大, 则浪费;
 - M 过小, 则将产生更多的溢出桶, 增加散列索引检索的时间。

➤ 动态散列索引

- ✓ 桶的数目随键值增多, 动态增加
- ✓ $h(k)$ 是和桶的数目 M 相关的。
 - M 的变化会否影响原来存储的内容呢?
 - 是否需要将原来已经散列-存储的数据按新的桶数重新进行散列-存储呢?



可扩展散列索引 (1)拟解决的问题

➤ 动态散列索引

可扩展散列索引

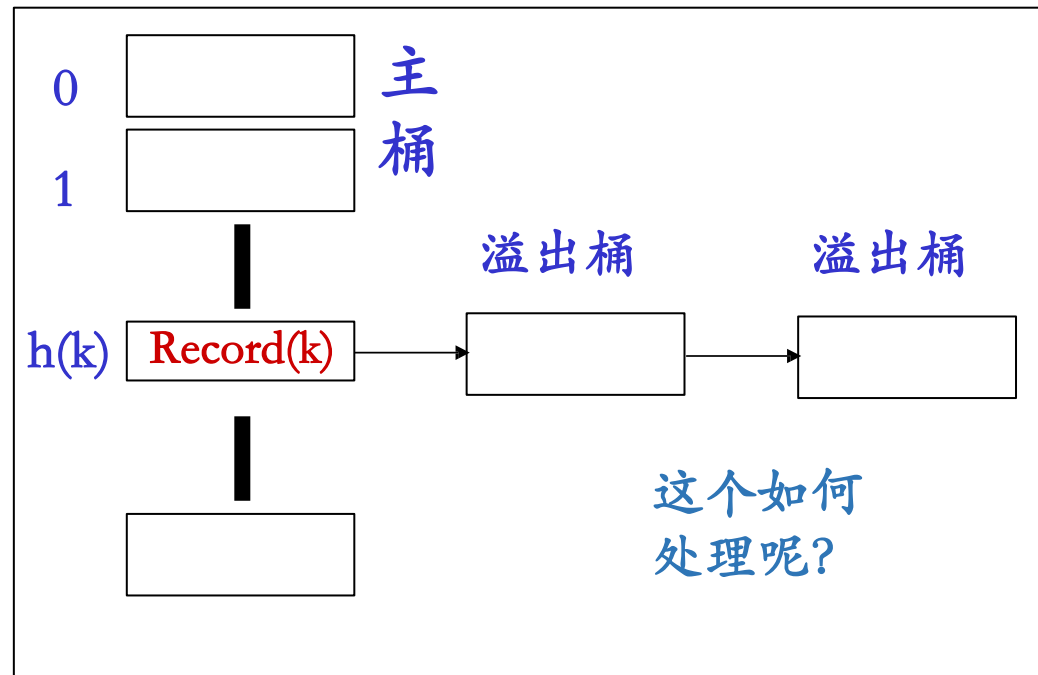
线性散列索引

✓ 桶的数目随键值增多，动态增加

✓ $h(k)$ 是和桶的数目 M 相关的。

M 的变化会否影响原来存储的内容呢？

是否需要将原来已经散列-存储的数据按新的桶数重新进行散列-存储呢？





可扩展散列索引 (2) 基本思想

➤ 可扩展散列索引

- ✓ 为桶引入一间接层，即用一个指向块的指针数组来表示桶，而不是用数据块本身组成的数组来表示桶
- ✓ 指针数组能增长，**其长度总是2的幂**。因而数组每增长一次，桶的数目就翻倍。不过，并非每个桶都有一个数据块；如果某些桶中的所有记录可以放在一个块中，则这些桶可能共享一个块。
- ✓ 散列函数 h 为每个键计算出一个 K 位二进制序列，该 K 足够大，比如32。但是桶的数目总是使用从序列第一位或最后一位算起的若干位，此位数小于 K ，比如说 i 位。也就是说，当 i 是使用的位数时，桶数组将有 2^i 个项。



可扩展散列索引 (3)基本思想示例

参数 k

参数 i

参数 $n=2^i$

i 为散列函数当前已经使用到的最多位数。即当前的桶数为 2^i 。

右上角标记本块散列函数使用位数。--局部

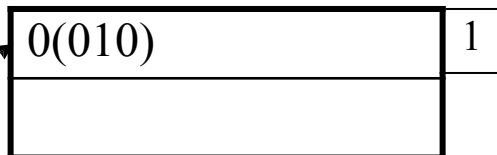
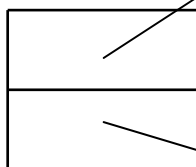
可变的桶
(指针数组)

由 2^0 个, 2^1 个, 2^2 个...
逐渐增加至 2^k 个

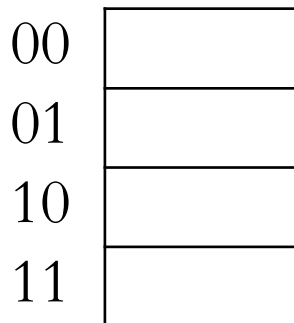
$i=1$

0

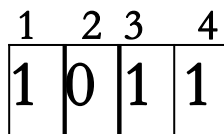
1



k 为散列函数所可能使用的最多位数。即可能的最大桶的数目为 2^k , 逐渐被使用。



i 的位次



k 位二进制位

索引块

取前 i 位, 按照前 i 位的值找到对应的索引块

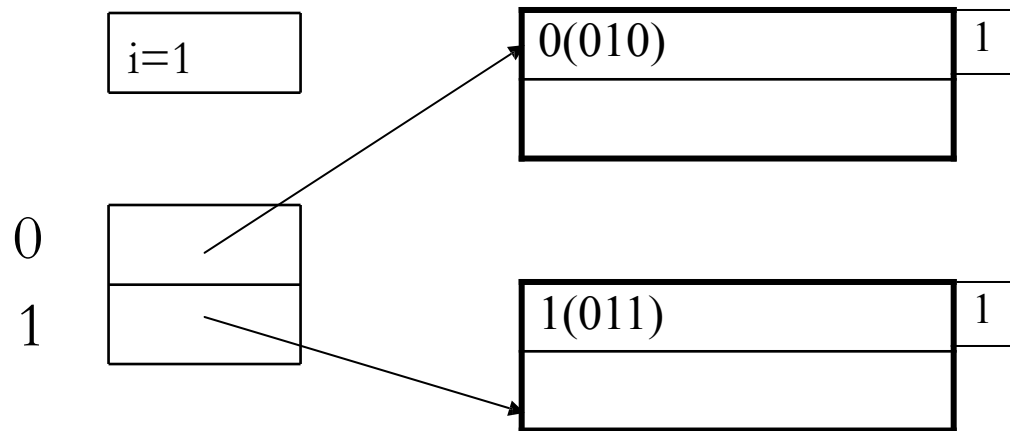


可扩展散列索引 (4)操作示例

$k=4$

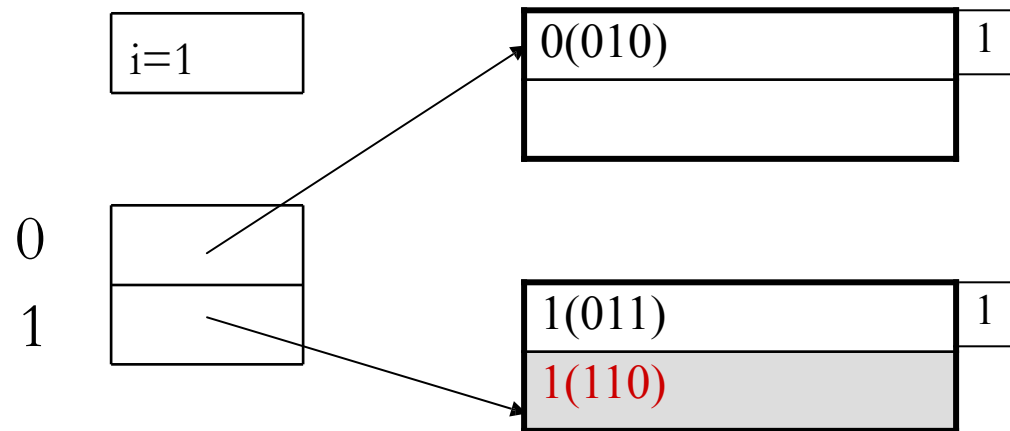
$i=1$

$n=2^i=2$



➤ 插入 **1**110

- ✓ 取前*i*位，确定索引块。
- ✓ 如有空间，则存储





可扩展散列索引 (4)操作示例

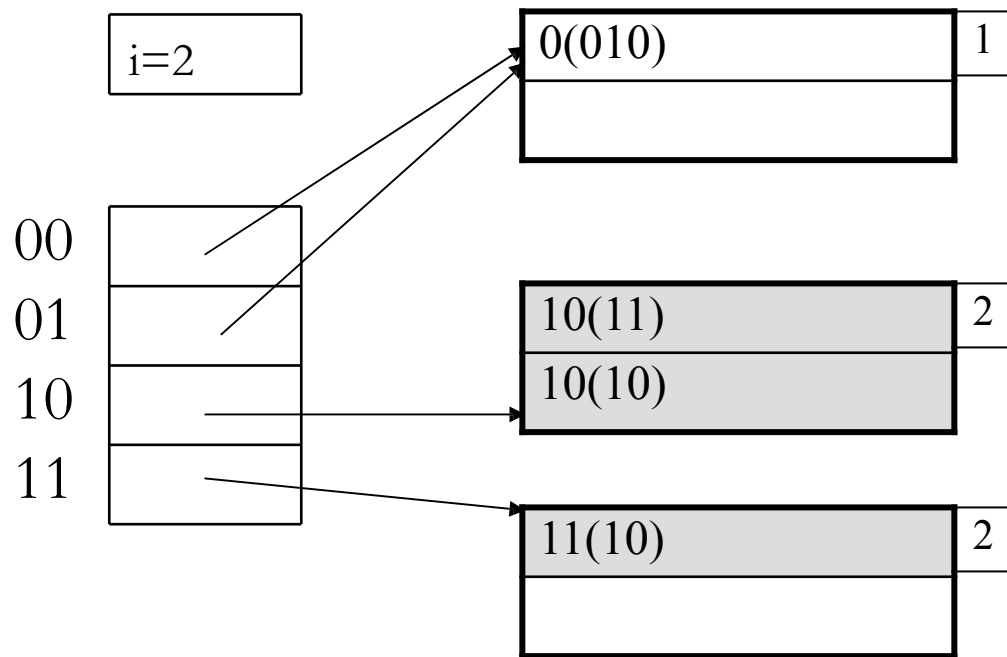
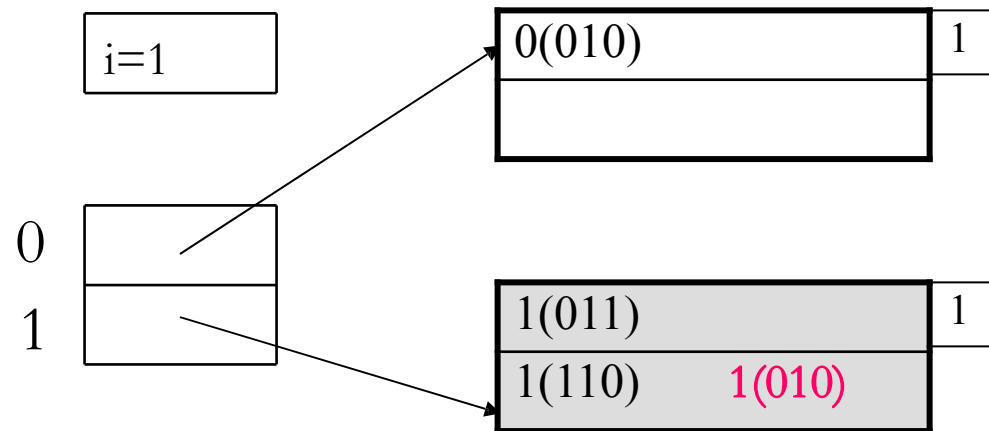
➤ 插入 1010

- ✓ 取前*i*位，确定索引块。
- ✓ 如已满，则需要扩展散列桶，进行分裂：
 - *i*增加1
 - 重新散列该块的数据到两个块中。其他不变

$k=4$

$i=2$

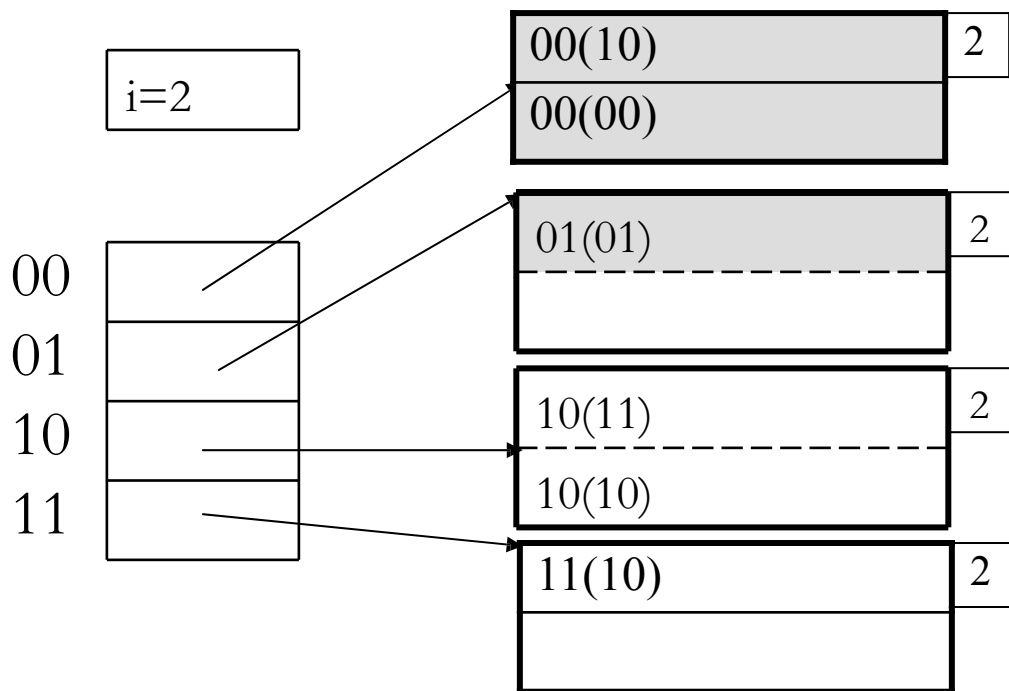
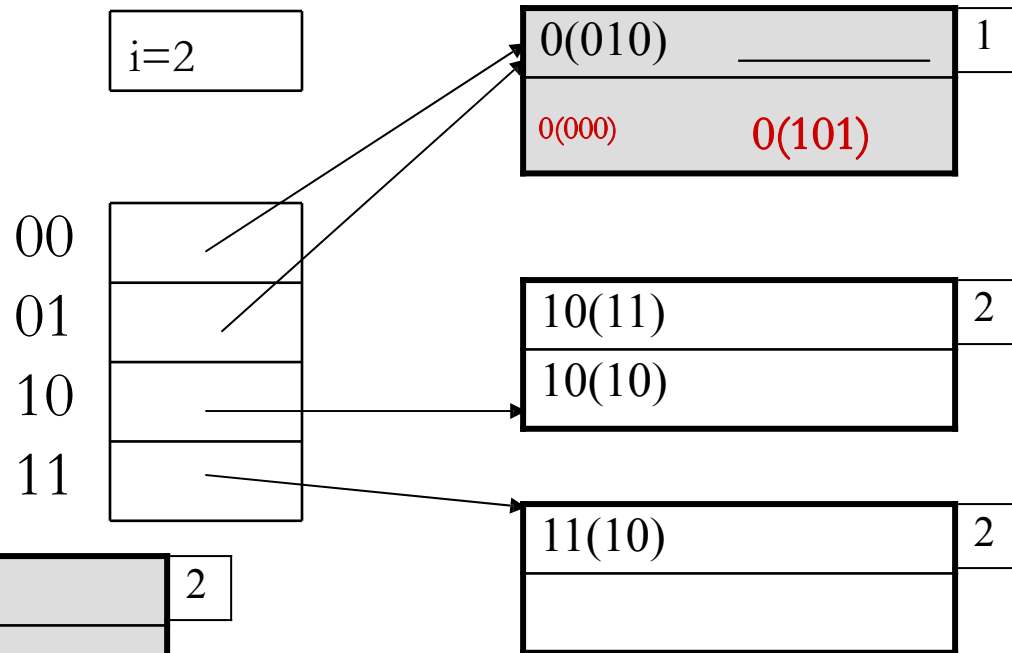
$n=2^i=4$





可扩展散列索引 (4)操作示例

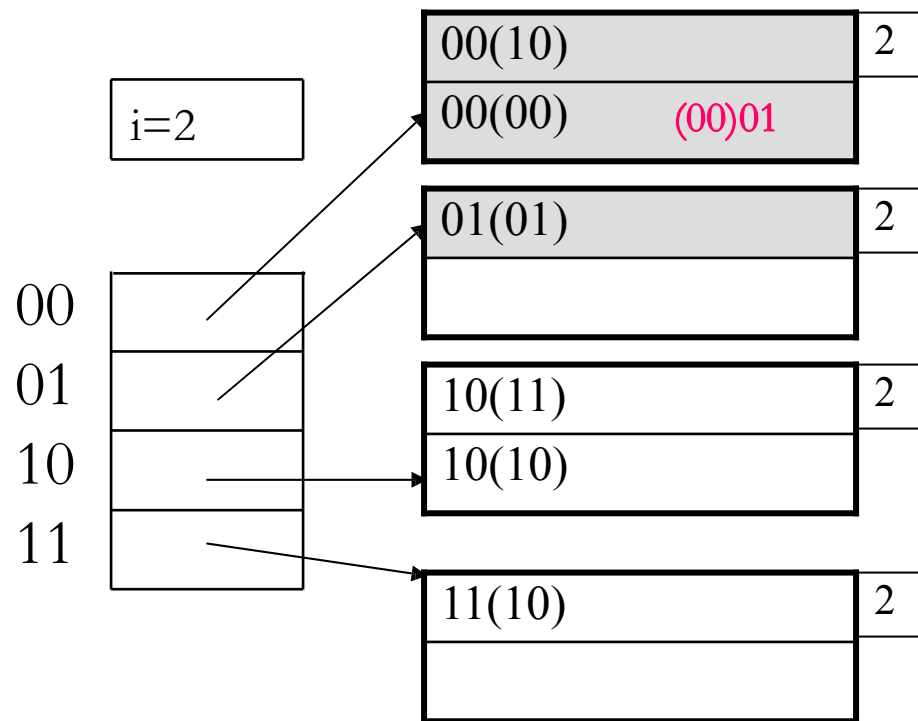
- 插入 0000
- 再插入 0101



- ✓ 取前*i*位，确定索引块。
- ✓ 如未滿，則存入。
- ✓ 如已滿，則需要擴展散列桶，進行分裂：
 - 暫時*i*可能無需增加
 - 重新散列該塊的數據到兩個塊中。其他不變

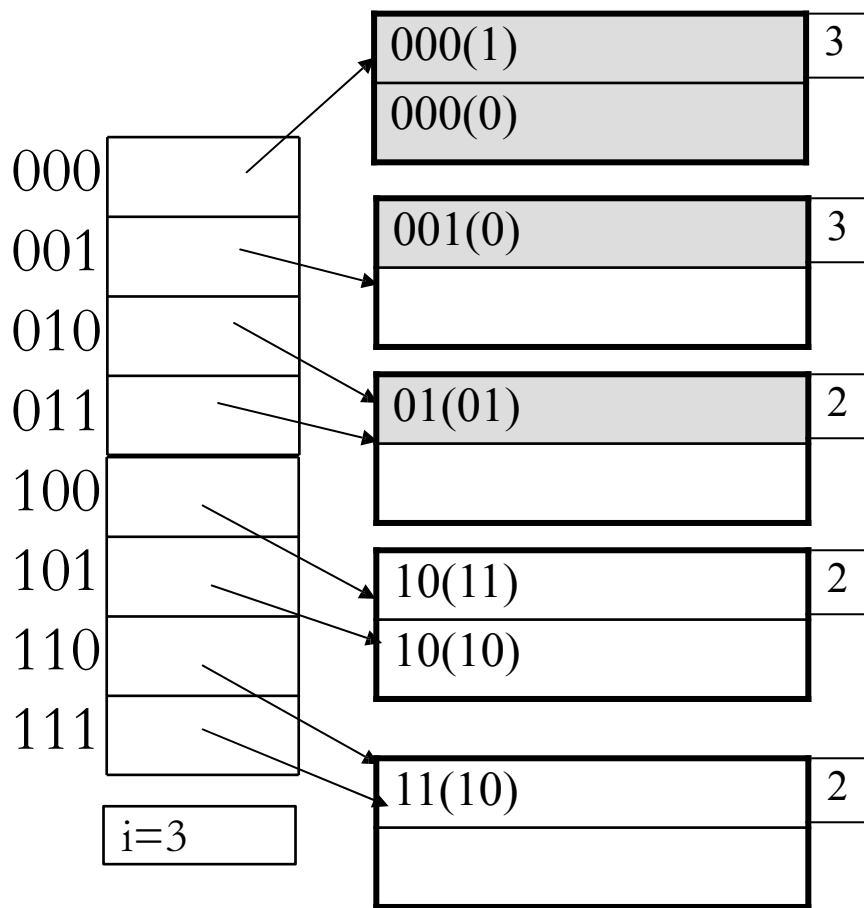


可扩展散列索引 (4)操作示例



- ✓ 取前 i 位，确定索引块。
- ✓ 如未滿，则存入。
- ✓ 如已滿，则需要扩展散列桶，进行分裂：
 - 此时 i 需增加1
 - 重新散列该块的数据到两个块中。其他不变

➤ 插入 0001





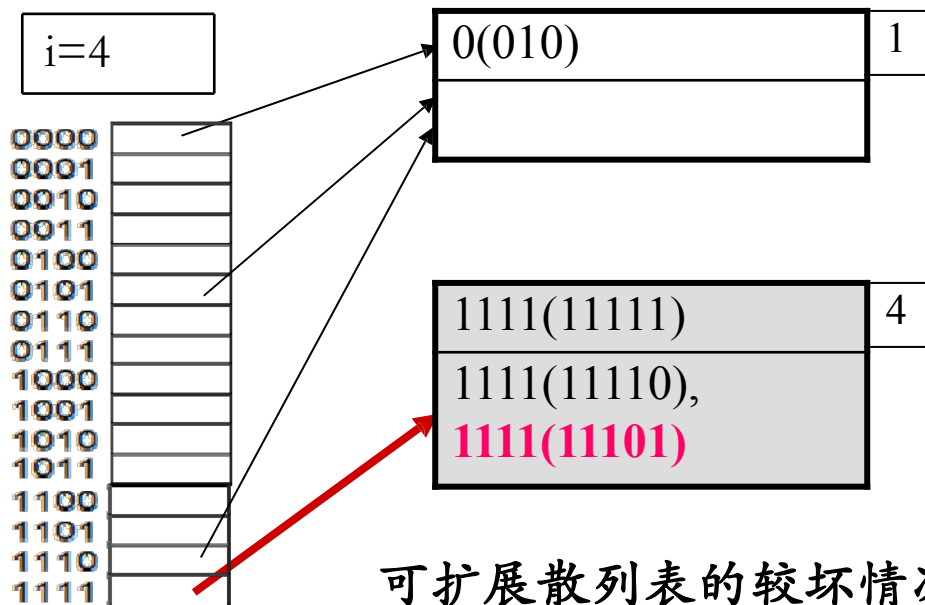
可扩展散列索引 (5)存在的问题

如何
解决?

➤ 问题

- ✓ 当桶数组需要翻倍时，要做大量的工作(当 i 很大时)；
- ✓ 当桶数翻倍后，其在主存中可能就装不下了，或者要占用更大的空间
- ✓ 如果每块的记录数很少，那么很有可能某一块的分裂比在逻辑上需要的分裂时间提前很多。例如：块中存放2个记录，即使记录总数远小于 2^{20} ，但也可能出现三个记录的前20位二进制位序列一样，在这种情况下，将不得不使用 $i=20$ 和100万个桶数组。

$i=1$ 时3个数组在一个桶中
；要分裂，则2个桶
 $i=2$ 时三个数组仍旧在一个桶中；要分裂，则 2^2 个桶
 $i=3$ 时三个数组仍旧在一个桶中；要分裂，则 2^3 个桶
... ..



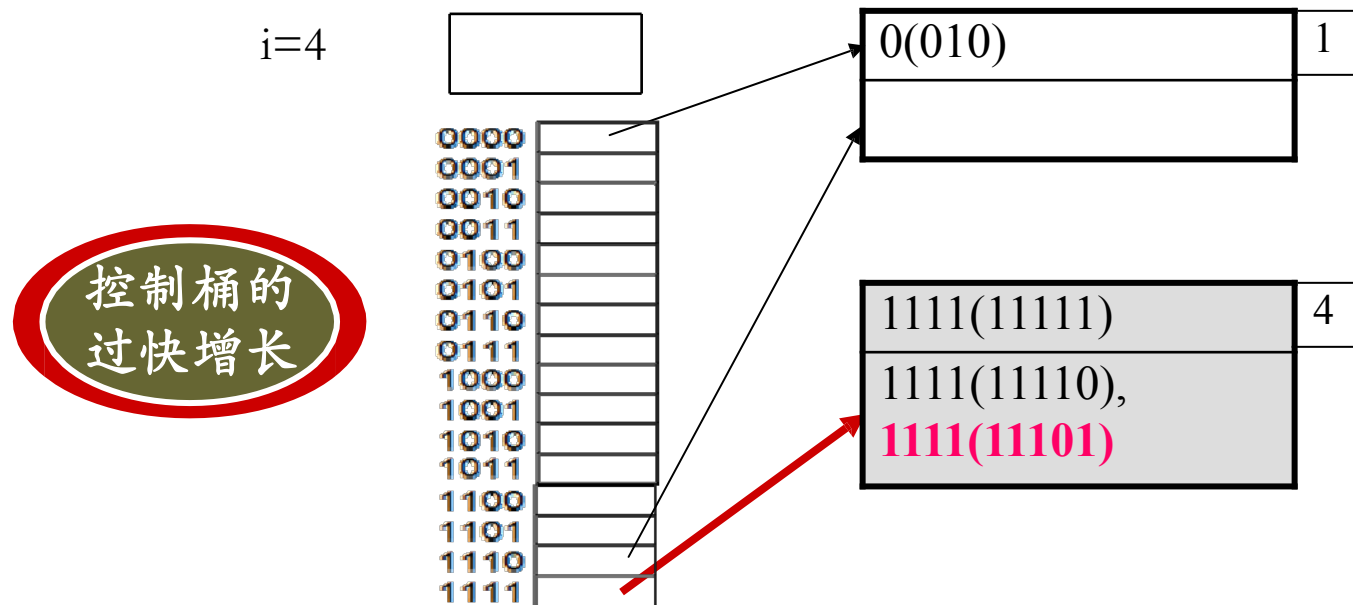
可扩展散列表的较坏情况



线性散列索引 (1)拟解决的问题

➤ 问题

- ✓ 桶的数目过快增长，但其**利用率不足**
- ✓ 例如：块中存放2个记录，即使记录总数远小于 2^{20} ，但也可能出现三个记录的前20位二进制位序列一样，在这种情况下，将不得不使用 $i=20$ 和100万个桶数组。但大部分桶都空闲未用。



可扩展散列表的较坏情况



线性散列索引 (2) 基本思想

➤ 线性散列索引

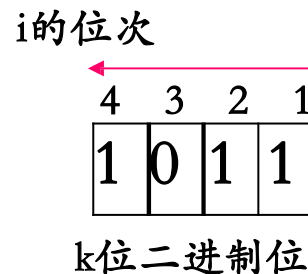
- ✓ 桶数 n 的选择：总是使存储块的平均记录数，保持与存储块所能容纳的记录总数成一个固定的比例，例如80%。超过此比例，则桶数增长1块，分裂。

---线性增长，每次增1。

- ✓ 存储块并不总是可以分裂，所以允许有溢出块，尽管每个桶的平均溢出块数远小于1。

- ✓ 用来做桶数组项序号的二进制位数是 $\lceil \log_2 n \rceil$ ，其中 n 是当前的桶数。

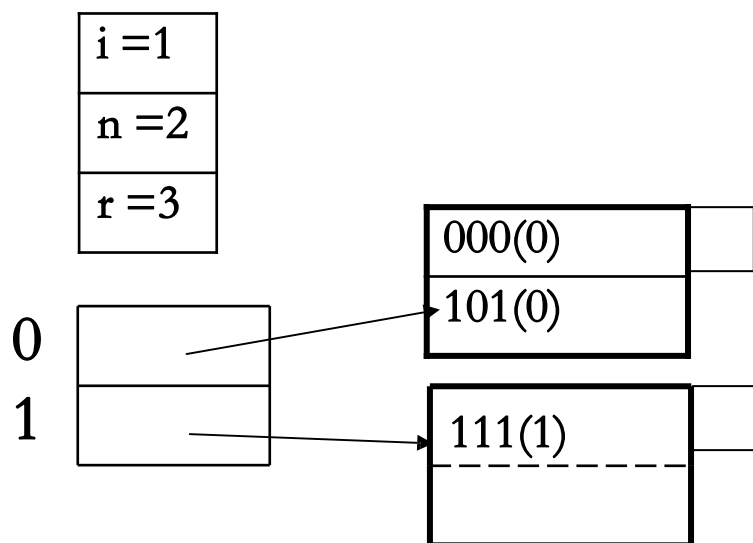
这些位总是从散列函数得到的位序列的右端(即低位)开始取。





线性散列索引 (2) 基本思想

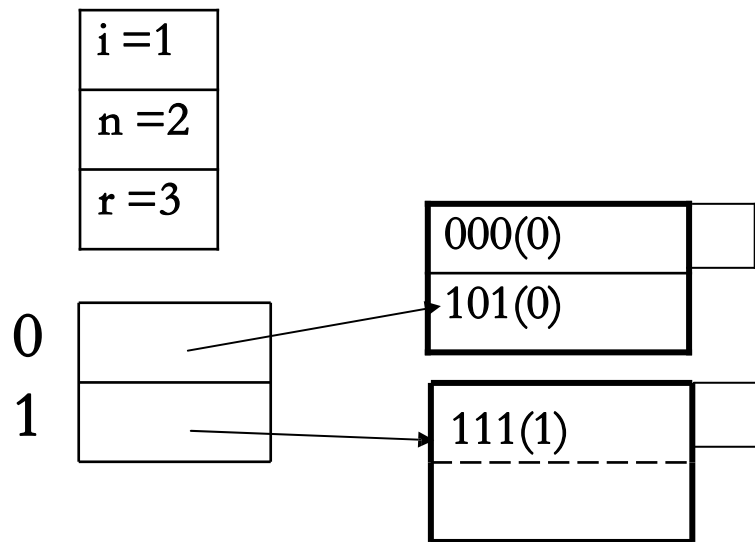
- ✓ 假定散列函数值的 i 位为桶数组项编号，且有一个键值为 K 的记录想要插入到编号为 $a_1a_2\cdots a_i$ 的桶中，即 $a_1a_2\cdots a_i$ 是 $h(K)$ 的后 i 位。把 $a_1a_2\cdots a_i$ 当作二进制整数，设它为 m 。 n 为当前桶数
- 如果 $m < n$ ，则编号为 m 的桶存在，并把记录存入该桶中。
 - 如果 $n \leq m < 2^i$ ，那么桶还不存在，因此我们把记录存入桶 $m - 2^{i-1}$ ，也就是当我们把 a_1 (它肯定是1) 改为0时对应的桶。



i -当前使用的散列函数的位数
 n -当前的桶数
 r -当前散列表中的记录总数
要求 $r \leq 1.7n$

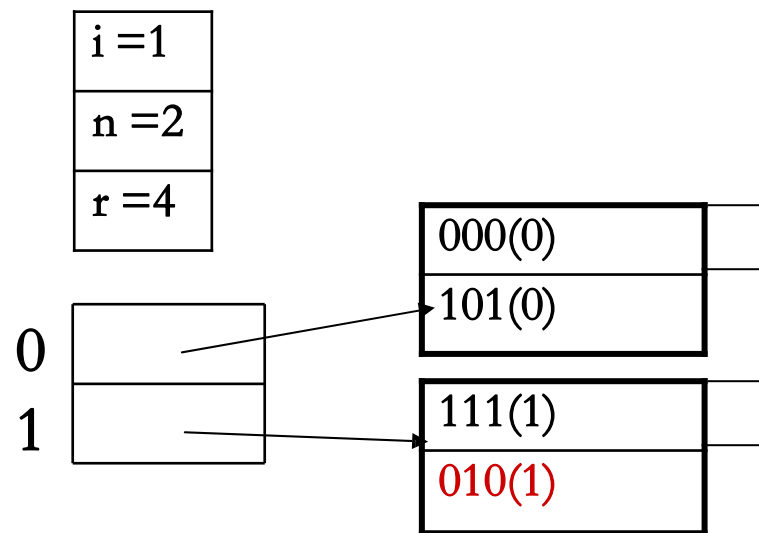


线性散列索引 (3)操作示例



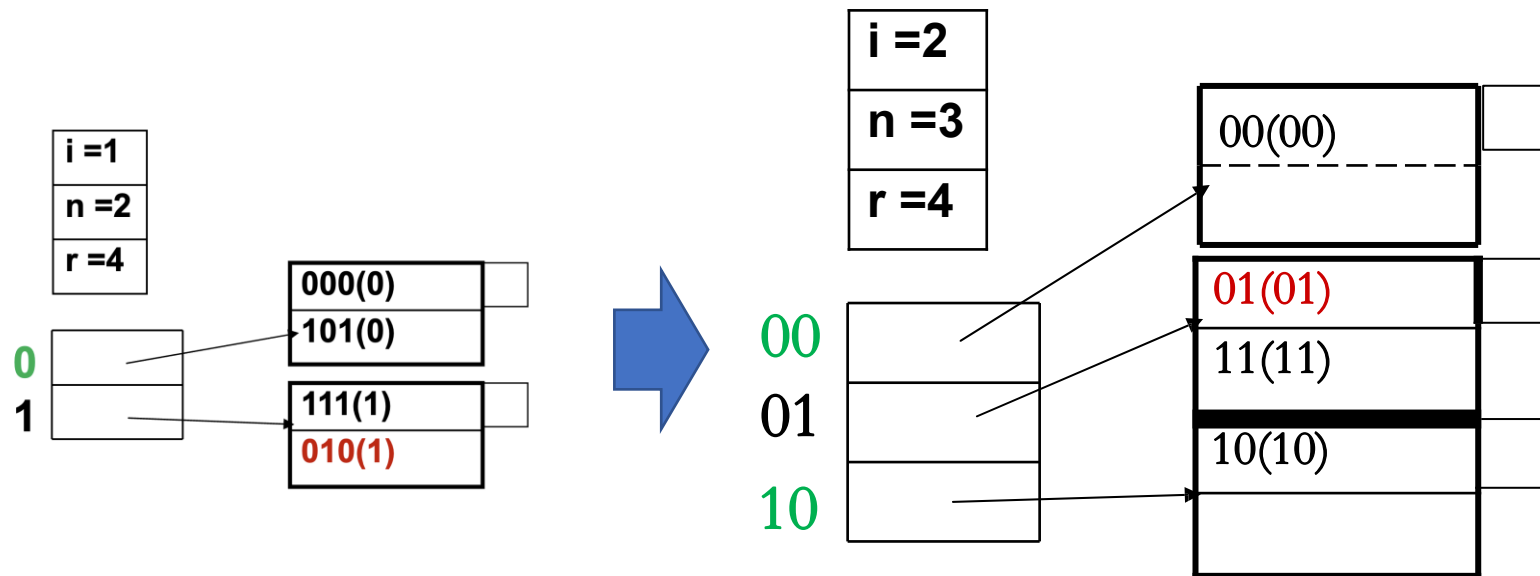
- ✓ 插入 0101 的记录, 如右图。
- ✓ 此时 $r/n = 4/2 = 2 > 1.7$, 桶数需要增1, 即要增加一桶, $n=3$
- ✓ 需要分裂。10桶要由00桶分裂而来(低位相同而最高位不同的那一个桶)

i -当前使用的散列函数的位数
 n -当前的桶数
 r -当前散列表中的记录总数
要求 $r \leq 1.7n$



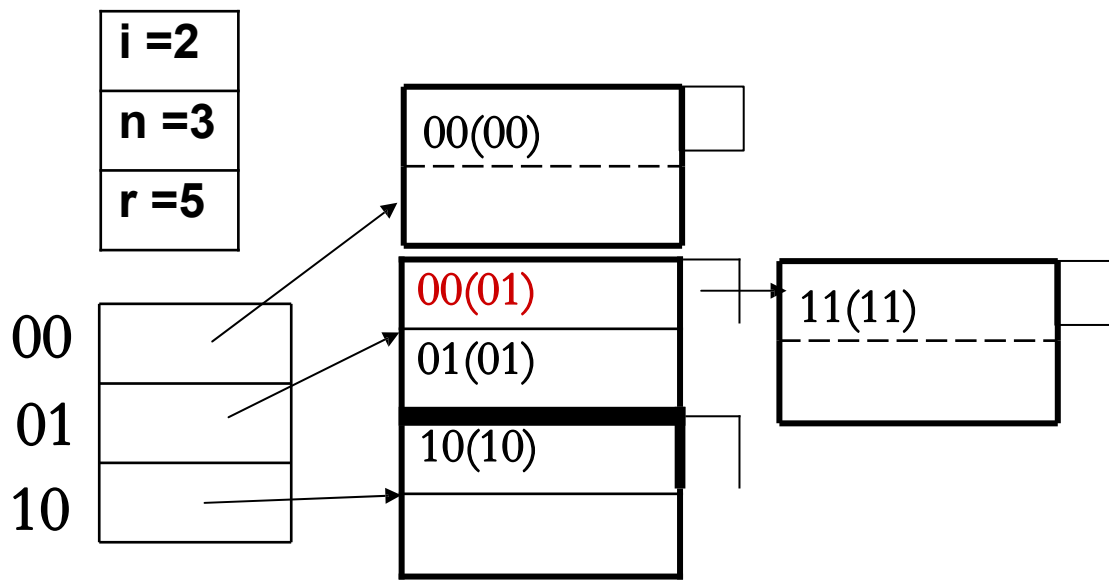


线性散列索引 (3)操作示例



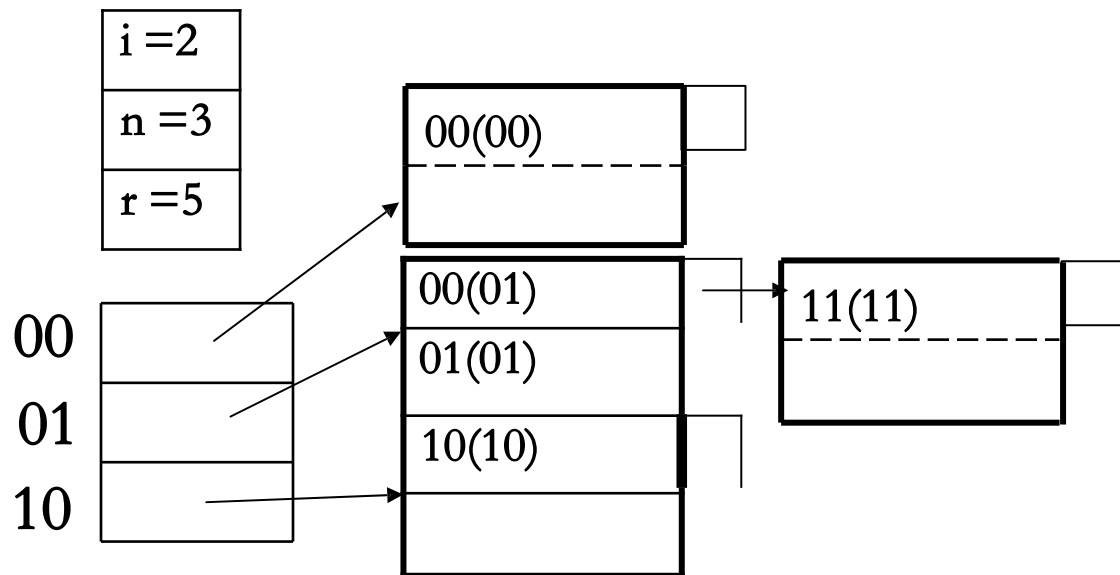
i-当前使用的散列函数的位数
n-当前的桶数
r-当前散列表中的记录总数
要求 $r \leq 1.7n$

- 插入 0001 的记录, 如右图。
- 此时 $r/n = 5/3 = 1.66 < 1.7$, 桶数无需增加



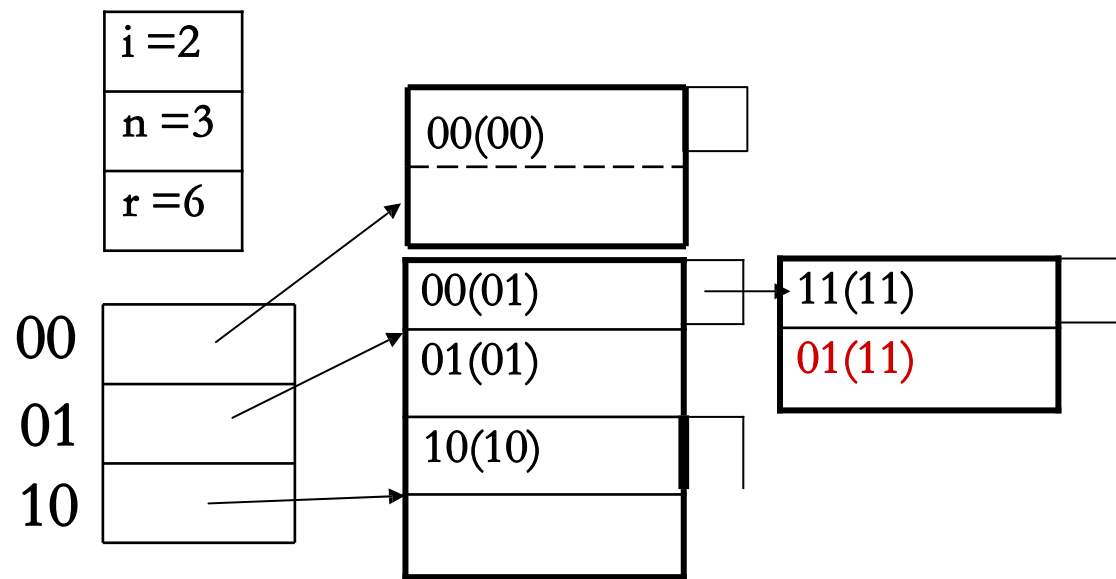


线性散列索引 (3)操作示例



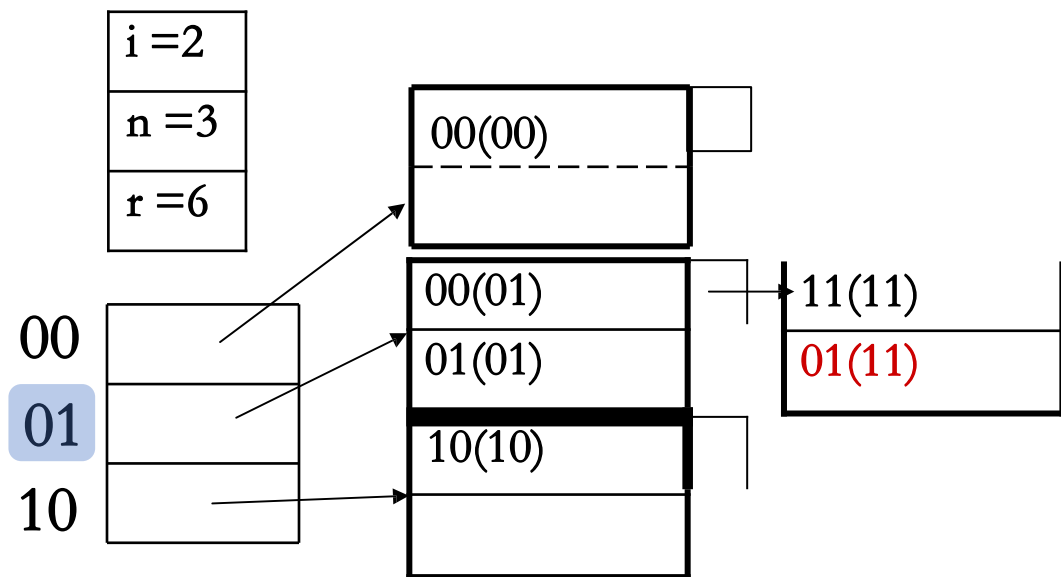
i -当前使用的散列函数的位数
 n -当前的桶数
 r -当前散列表中的记录总数
要求 $r \leq 1.7n$

- ✓ 插入 0111 的记录, 如右图。
- ✓ 因没有 11 桶, 则插入 01 桶。
- ✓ 此时 $r/n = 6/3 = 2 > 1.7$, 桶数需要增 1, 即要增加一桶 11, n 仍然为 3
- ✓ 需要分裂。11 要由 01 桶分裂而来(低位相同而最高位不同的那一个桶)



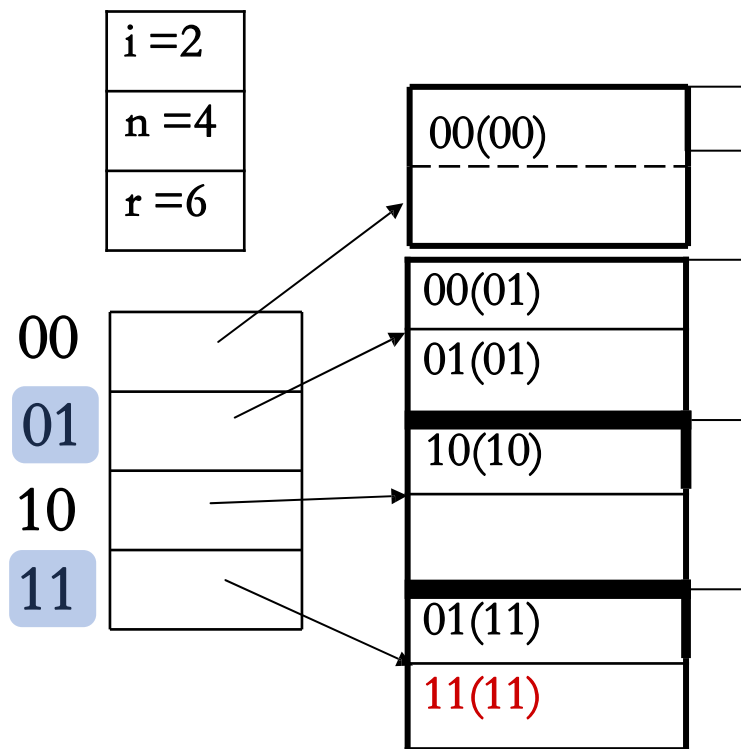


线性散列索引 (3)操作示例



分裂前

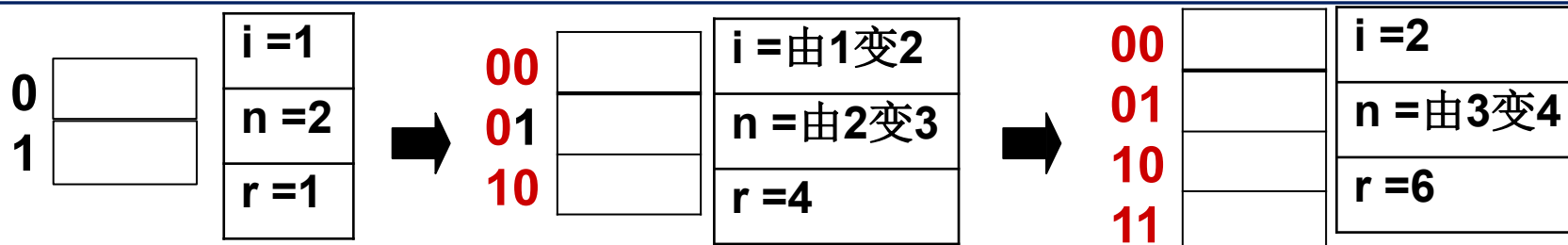
分裂后



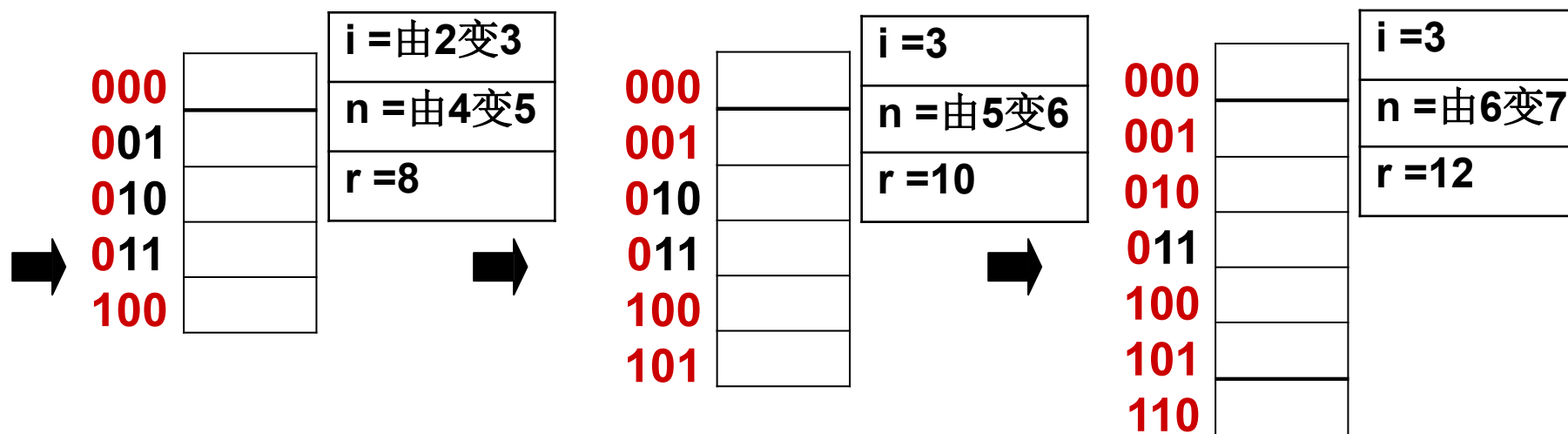
i -当前使用的散列函数的位数
 n -当前的桶数
 r -当前散列表中的记录总数
要求 $r \leq 1.7n$



线性散列索引 (4)回顾操作特点



- 键值由低位向高位次序使用。对应索引项存储在由桶数组指针指向的存储块
- 当 r/n (当前索引项数除以桶数) 大于一定比例时, 按次序增加1个桶
- 假设该桶为 $1a_2a_3...a_i$, 则该桶由 $0a_2a_3...a_i$ 对应的块分裂而来
- 当桶数超过 2^i 个桶时, 则使 i 增1。





回顾本讲学习了什么?

