Database HW 5

PB18111697 王章瀚

1.

1、假设我们在数据库中设计了如下基本表来存储文献：paper(id: int, title: varchar(200), abstract: varchar(1000))。最常见的文献查询可以描述为"查询 title 中同时包含给定关键词的文献"，关键词可以是一个，也可以是多个。请回答下面问题（假设所有文献都是英文文献）：

1) 假如在 title 上创建了 B+-tree 索引，能不能提高此查询的效率（须解释理由）？

2) 由于文献 title 的关键词中存在很多重复词语，因此上述文献查询可以借鉴我们课上讲述的支持重复键值的辅助索引技术来进一步优化。请基于此思想画出一种优化的索引结构，简要说明该索引上的记录插入过程以及文献查询过程。

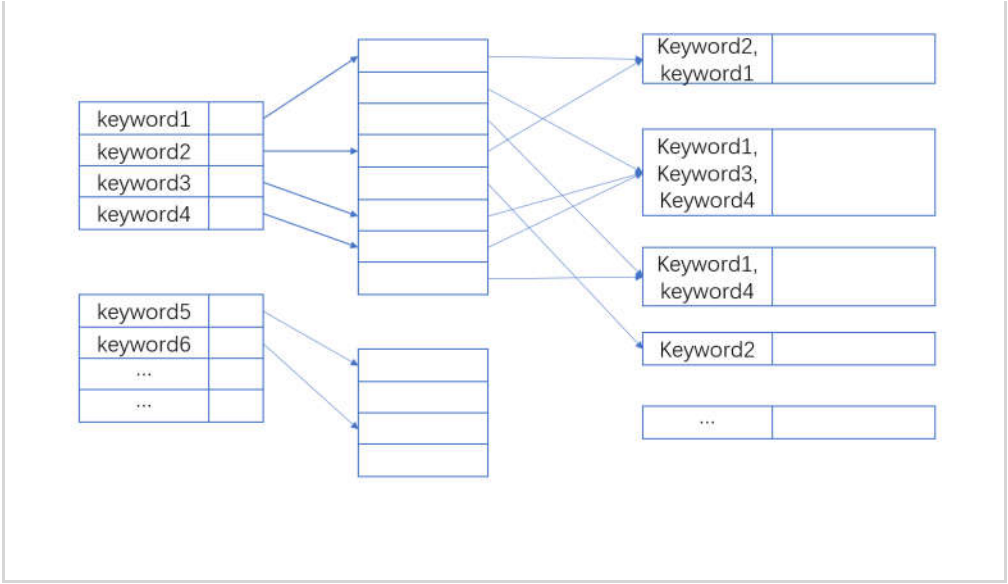
1).

不能利用在 title 上建 B+-Tree 来提升查询效率. B+ 树能提升查询效率是因为可以将查询的 input 按其序关系在树中逐层查询. 而此处输入的关键词的词本身及其数量都不确定, 很难给出一个性能优异的编码使得这样的查询能够直接作用在 title 上建立的 B+ 树索引上. 另一方面, 关键词和 title 之间的相对复杂的映射关系也使得 title 上的索引对于查关键词来说作用不那么大.

2).

构建一个带间接桶辅助索引. 对于每一篇 paper,

- 插入的时候就将其title内所有关键词提取出来, 并从相应关键词的桶里指向该 paper.
- 查询的时候只需要将用户给出的关键词相应的桶都取出, 然后取一个交集即可.
- 索引表扩展等问题同正常索引.



2.

2. 假设有如下的键值，现用 5 位二进制序列来表示每个键值的 hash 值。回答问题：

A [11001] B [00111] C [00101] D [00110] E [10100] F [01000] G [00011]

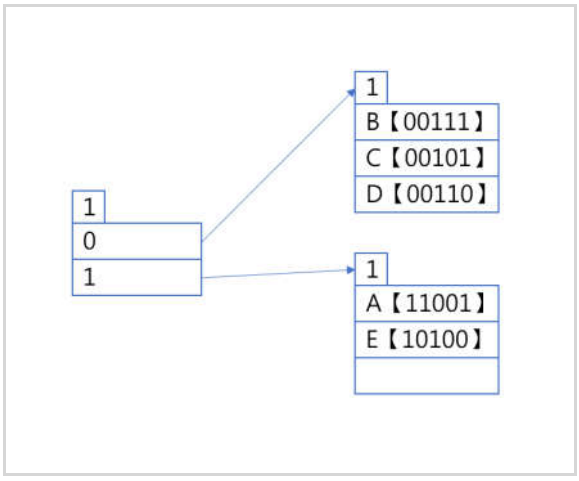
H [11110] I [10001] J [01101] K [10101] L [11100] M [01100] N [11111]

1）如果将上述键值按 A 到 N 的顺序插入到可扩展散列索引中，若每个桶大小为一个磁盘块，每个磁盘块最多可容纳 3 个键值，且初始时散列索引为空，则全部键值插入完成后该散列索引中共有几个桶？并请写出键值 E 所在的桶中的全部键值。

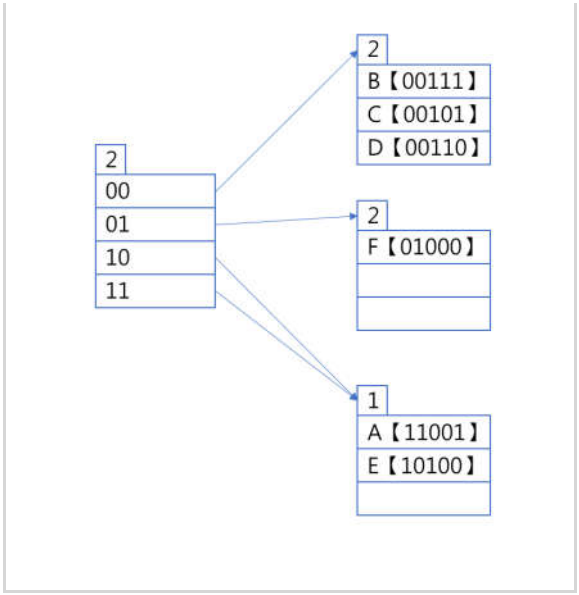
2）前一问题中，如果换成线性散列索引，其余假设不变，同时假设只有当插入新键值后空间利用率大于 80%时才增加新的桶，则全部键值按序插入完成后该散列索引中共有几个桶？并请写出键值 B 所在的桶中的全部键值（包括溢出块中的键值）。

1).

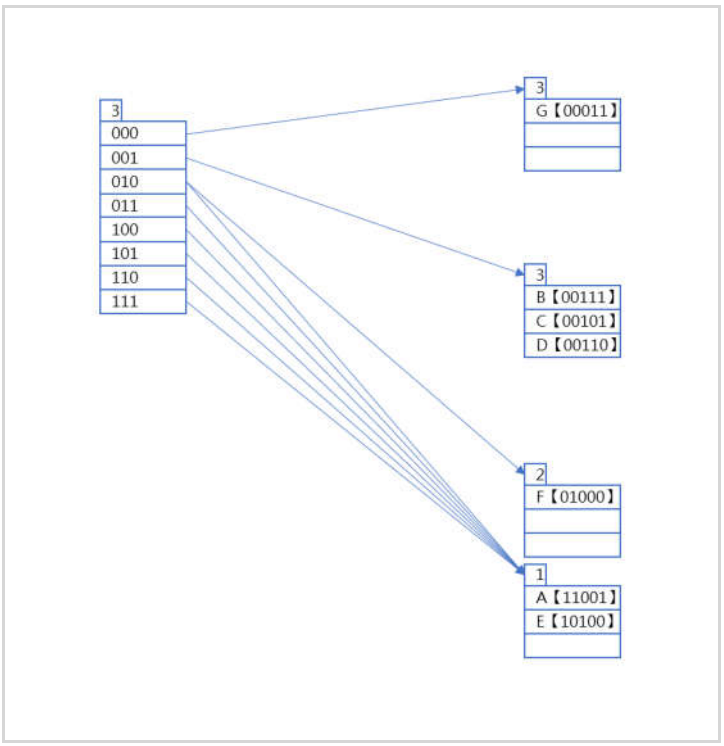
每个桶可以有三个键值. 不妨假设刚开始以最高 1 位表示桶, 则插入 A, B, C, D, E 后有:



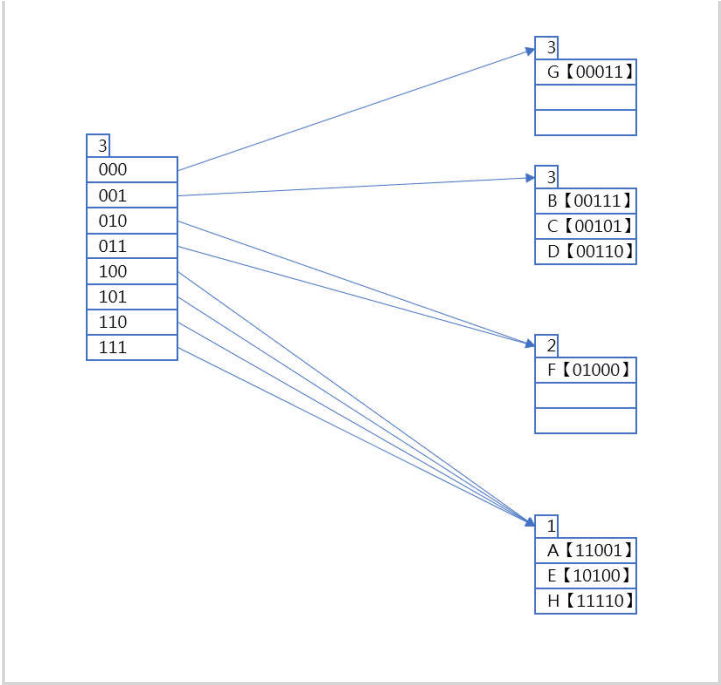
插入 F 的时候需要扩展, 如下:



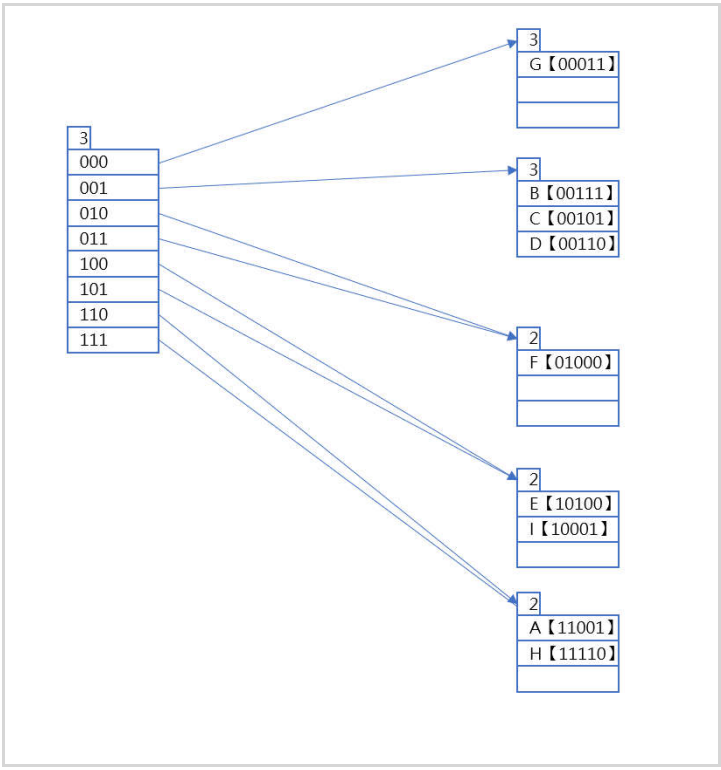
插入 G 的时候需要扩展, 如下:



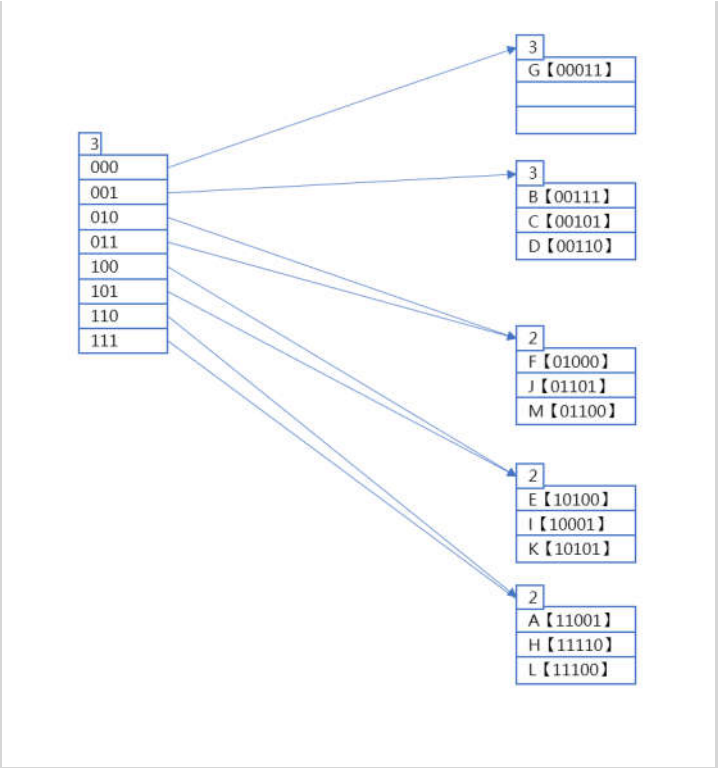
插入 H, 如下:



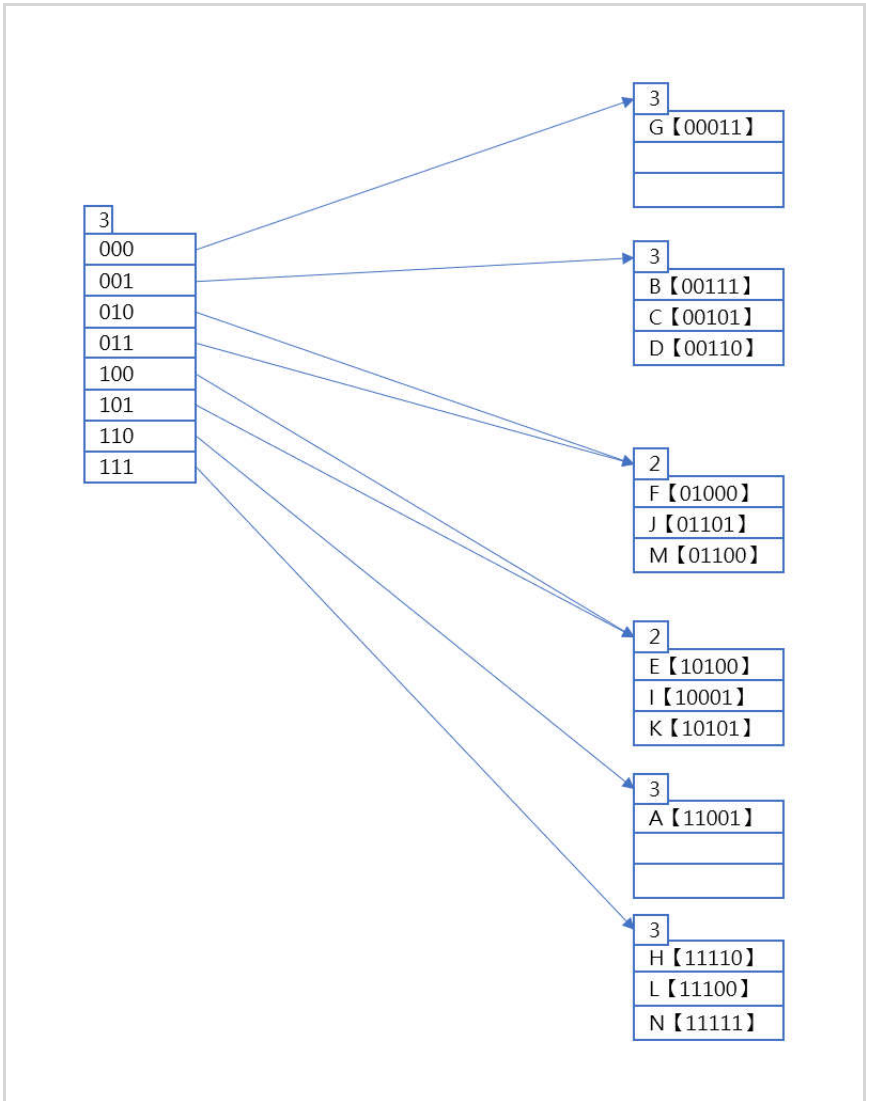
插入 I 需要扩展, 如下:



插入 J, K, L, M, 如下:



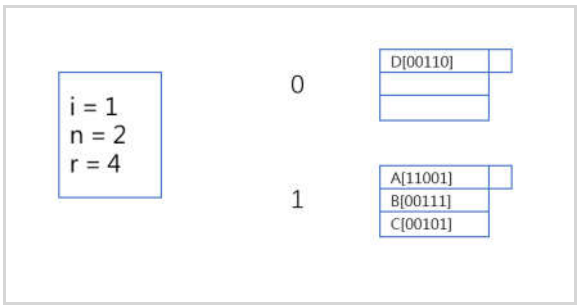
插入 N 需要扩展, 如下:



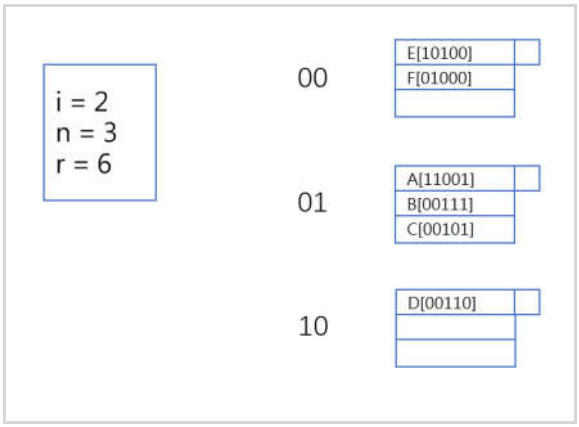
因此最后有 6 个桶, 键值 E【10100】 所在桶全部键值为: E【10100】, I【10001】, K【10101】.

2).

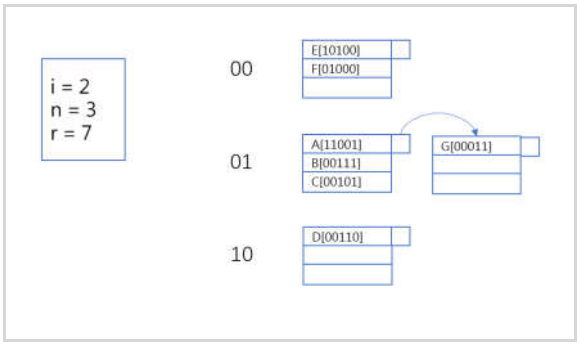
插入 A, B, C, D 之后, 得到如下:



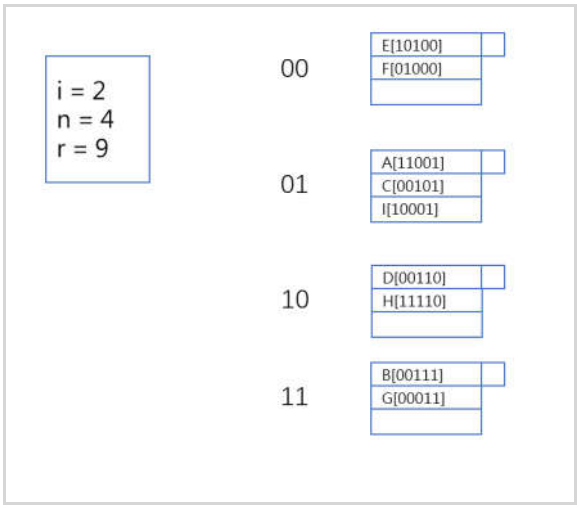
插入 E 之后 $\frac{5}{3*2} > 80\%$, 应当扩展, 然后还能继续插入 F, 得到如下:



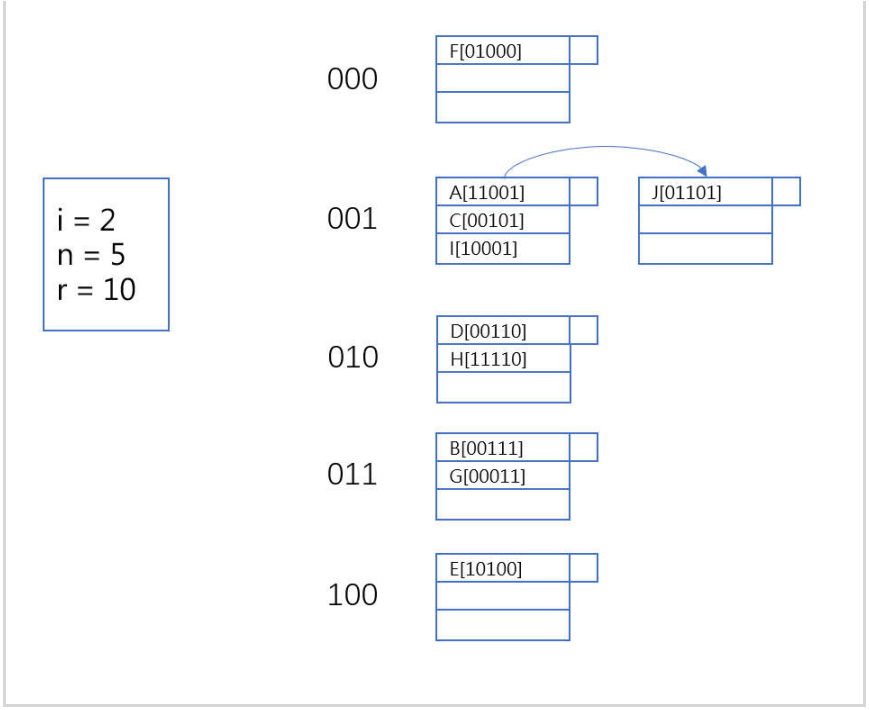
插入 G 的时候, 需要溢出块:



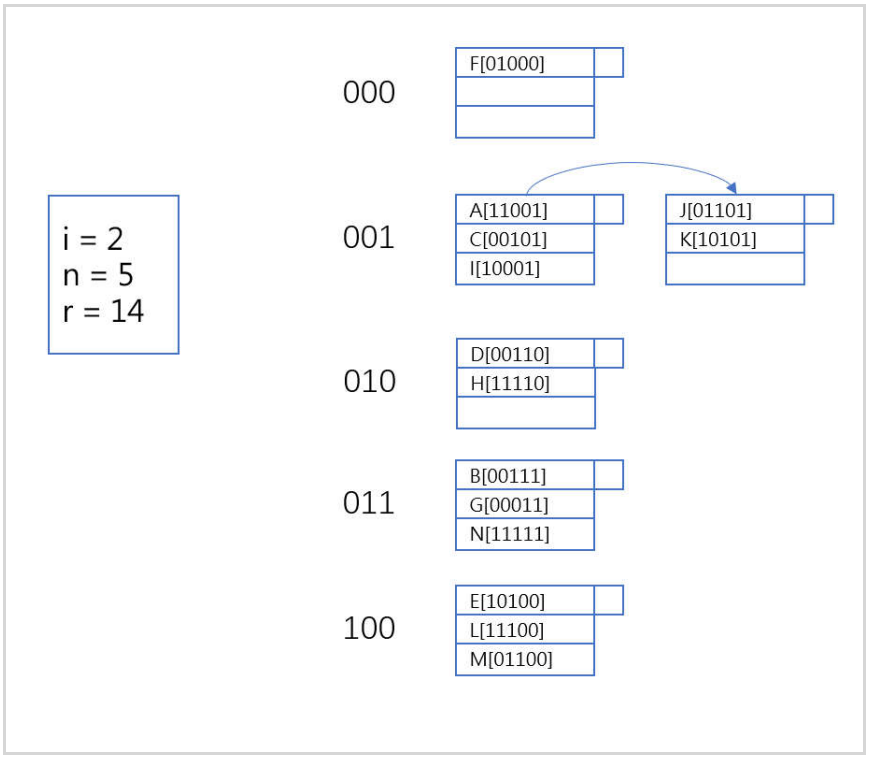
插入 H 的时候, 需要扩展, 然后可以直接插入 I:



插入 J 需要扩展:



插入 K, L, M, N 都可以直接插入:



因此共有 5 个桶(溢出块不算桶). 键值 $B[00111]$ 所在得桶全部键值为:
$B[00111], G[00011], N[11111]$

3.

3、对于 B+树，假设有以下的参数：

参数	含义	参数	含义
N	记录数	*S*	读取一个磁盘块时的寻道时间
n	B+树的阶，即节点能容纳的键数	*T*	读取一个磁盘块时的传输时间
R	读取一个磁盘块时的旋转延迟	*m*	在内存的 *m* 条记录中查找 1 条记录的时间（线性查找）

假设所有磁盘块都不在内存中。现在我们考虑一种压缩 B+树，即对 B+树的节点键值进行压缩存储。假设每个节点中的键值压缩 1 倍，即每个节点在满的情况下可压缩存储 2*n* 个压缩前的键值和 2*n*+1 个指针。额外代价是记录读入内存后必须解压，设每个压缩键值的内存解压时间为 *c*。给定 N 条记录，现使用压缩 B+树进行索引，请问在一棵满的 *n* 阶压缩 B+树中查找给定记录地址的时间是多少？（使用表格中的参数表示，*n*+1 或 *n*–1 可近似表示为 *n*）？

在此题中, 读取一个磁盘块所需的时间是 $R+S+T$. 假设一个磁盘块恰好能够存储一个B+ Tree 节点.

那么对于 n 阶压缩 B+ 树, 它存储 N 个记录最多需要约 $\log_{2n} N$ 层,

每层中查找时, 只会针对一个块内进行遍历, 而每次都要调入块:

- 每块调入需要 $R+S+T$
- 每块遍历找键值需要 $2n \times c = 2cn$

因此查找给定记录的地址时间是: $(R+S+T+2cn)\log 2nN$