ADB Sheet 2 Sol.

14.14)

B = 512 byte
P = 6 bytes (block Pointer)
PR = 7 bytes (record Pointer)

$r = 30\times10^3$

Name	SSN	Dept Code	Addr.	Phone	Bdate	Sex	Job Code	Salary	// del
30 byte	9 byte	9 B	40B	9B	8B	1B	4B	4B	1B

a) Record length (R)

$R = 30+9+9+40+9+8+1+4+4+1 = 115\,B$

b) blocking factor (bfr) and no. of blocks (b) (unspanned)

$bfr = \lfloor B/R \rfloor = \lfloor 512/115 \rfloor = 4$ record/block

$b = \lceil r/bfr \rceil = \lceil 30\times10^3/4 \rceil = 7500$ blocks

c) Suppose file is ordered by key field SSN and we want to construct a Primary index on it

i) index blocking factor bfr$_i$

$$R_{ind} = \underset{\substack{\uparrow \\ \text{SSN} \\ \text{(index Key)}}}{9} + \underset{\substack{\uparrow \\ \text{block} \\ \text{Pointer}}}{6} = 15$$

$bfr_i = \lceil \frac{B}{R_i} \rceil = \lceil 512/15 \rceil = 34$ index records/block

ii) The no. of first-level entries and no. of 1st level blocks (assuming one level)

$r_i = \underbrace{\#\text{ blocks in file}}_{b} = 7500$

- Primary index has row for each data block

$b_i = \lceil r_i / bfr_i \rceil = \lceil 7500/34 \rceil$

$= 221$ blocks

- the index itself will span 221 blocks

iii) • We keep making levels until the last level has just 1 block (how many levels for a multi-level index)

- We have

$r_{i1} = 7500$ $\quad$ $b_{i1} = 221$ block

- by creating a primary index on top of that:

$r_{i2} = 221$ entries $b_{i2} = \lceil r_{i2} / bfr_i \rceil = 7$ blocks

Justification:

→ the 2nd level has an entry for each block in the 1st level

→ bfr is the same because R_i is still SSN and pointer

- Primary index on top of that:

$r_{i3} = 7$ entries $\quad$ $b_{i3} = \lceil 7/34 \rceil = 1$

#stop

→ index has 3 levels $(X = 3)$

- Notice that $\log_m m^K$ is mathematically the no. of times we need to divide m^K by m to reach 1

→ Can hence compute no. of levels directly as $\lceil \log_{bfr} r_{i1} \rceil$

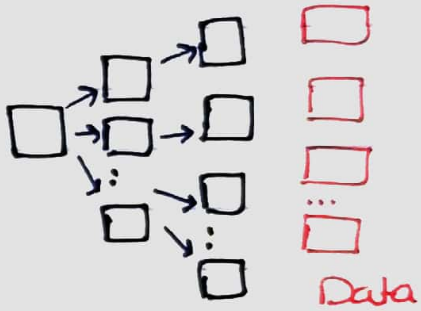
.or less if we ceil the log and $N \neq m^K$

iv) the total no. of blocks required by multi-level index

$$\rightarrow \sum_j b_{ij} = b_{i1} + b_{i2} + b_{i3} = 221 + 7 + 1$$
$$= 229 \text{ blocks}$$

v) the no. of block IOs to search for a record given SSN

$$\#IO = \underset{\substack{\uparrow \\ \text{no. of levels}}}{X} + 1 = 4$$



$$\underset{\substack{\text{access} \\ \text{top level}}}{1} + 1 + 1 + \underset{\substack{\text{access} \\ \text{data}}}{1}$$

d) Suppose file is not ordered by key field SSN and we want to construct a secondary index on it

- recall that a secondary index (key) is dense (has a row for each data record (where the pointer points to the block where the record exists).

i) bfr_i

$$R_i = 9 + 6 = 15$$
$$bfr_i = \lfloor B/R_i \rfloor = 34 \text{ index records/block}$$

ii) r_i and b_i (1st level entries and blocks)

$$r_i = r = 30 \times 10^3$$

$b_i = \lceil r_i / bfr_i \rceil = \lceil 30\times10^3 / 34 \rceil = 883$ blocks

→ Notice that although index has a row for each record $b_i < b$ as $R_i < R$

iii) the no. of levels to make it multi-level

1st level:

$r_{i1} = 30\times10^3$ $b_{i1} = 883$ blocks

2nd level:

//its a Primary index on top of 1st level

•index is sorted & unique

$r_{i2} = 883$ $b_{i2} = \lceil r_{i2} / bfr_i \rceil = 26$

3rd level:

$r_{i3} = 26$ $b_{i3} = \lceil r_{i3} / bfr_i \rceil = 1$

•only need 3 levels

→ indeed $\lceil \log_{34}(30\times10^3) \rceil = 3$

iv) total no. of multi-level index blocks

$$\sum_j b_{ij} = 883 + 26 + 1 = 910$$

v) the no. of IOs to retrieve SSN (Search)

$$\#\text{IOs} = \underset{\substack{\uparrow \\ \text{no. of} \\ \text{levels}}}{X} + 1 = 4$$

(just like in the last example)

e) Suppose the file is unordered by the non-Key field deptCode and we want to create a Secondary index on it.

→ In this case, the index has a row for each unique value and each points to a block of pointers with record pointers to the data records of that value.

→ given as well is that there are 1000 distinct values of deptCode
.and that dept codes are evenly distributed among them ①

i) bfr_i

DeptCode

block Pointer Points to block of records Pointer

$$R_i = 9 + 6 = 15$$

$$bfr_i = \lfloor B / R_i \rfloor = 34 \text{ index record / block}$$

iii) the no. of 1st level index entries and blocks

$$r_i = \#\text{distinct values} = 1000 \text{ entry}$$

$$b_i = \lceil r_i / bfr_i \rceil = 30 \text{ block}$$

ii) the no. of blocks needed by level of indirection that stores record pointers

→ there's on average $\frac{r}{1000} = 30$ record having ① the same dept_code value

* i.e., each index entry (sorted unique values) needs 30 pointers to records

→ record pointer is 7 bytes, hence $30 \times 7 = 210$ byte < 512 byte and they all can fit in one block.

thus, each index entry will be associated with 1 block of Pointers

. implies 1000 block of Pointers in total

iv) the no. of levels X to make it multi-level

1st level

$r_{i1} = 1000$ $\qquad$ $b_{i1} = 30$

• Primary index on top

2nd level

$r_{i2} = 30$ $\qquad$ $b_{i2} = \lceil r_{i2} / bfr_i \rceil$

$= 1$

. hence 2 levels

• indeed $\lceil \log_{34} 1000 \rceil = 2$

→ this doesn't count the level of indirection

v) Total no. of blocks needed

$$\sum_j b_{ij} + b_{indir} = \underset{30}{b_{i1}} + \underset{1}{b_{i2}} + 1000$$

$$= 1031 \text{ block}$$

vi) Approx. no. of block IOs for exact search on dept_code

block of Ptrs has 30 Ptrs (ii)

$$\# IOs = \underbrace{X}_{\text{to reach level 1}} + \underbrace{1}_{\text{to reach block of Ptrs}} + \underbrace{30}_{\text{will read 30 data blocks}} = 33$$

f) Suppose file is ordered by non-Key field Dept_Code

- Construct a clustering index that uses block anchors

  i.e., every new value of dept Code starts in a new block (2nd version in lecture)

- assume 1000 distinct dept_Code values (evenly distributed)

→ In this case

- each distinct value is entry in index
- each points to a block for that value (that may point to another one).

i) bfr_i (index fanout)

$$R_i = 9+6 = 15 \text{ bytes}$$

$$bfr_i = \lceil 512/15 \rceil = 34 \text{ index / block}$$

ii) r_i and b_i

$$r_i = \# \text{distinct} = 1000 \quad \text{index entry}$$
$$b_i = \lceil r_i / bfr_i \rceil = 30 \text{ block}$$

iii) X (levels for multi-index)

1st level:

$r_{i1} = 1000$ $\qquad$ $b_{i1} = 30$

2nd level:

$r_{i2} = 30$ $\qquad$ $b_{i2} = \lceil r_i / bfr_i \rceil = 1$

- X=2

iv) total no. of blocks for multi-level ind.

$$\sum_j b_{ij} = b_{i1} + b_{i2} = 31$$

V) #IOs for exact Search by dePtCode

$\# IOs = X + 1 = 2 + 1$. this will give us the 1st block where records match

bfr for file is 4 ← is there more?

- 30 records for each distinct value means → $\lceil 30/4 \rceil = 8$ blocks in total

i.e., will need to access 7 more blocks

- assuming blocks are contiguous
- also if connected by Ptrs

$bfr = \lfloor \frac{512-6}{115} \rfloor = 4$

* Note that for Primary / Secondary (Key) index, its always one block as values are distinct.

* For Secondary (Non-Key), block of Pointers handle that.

- Note that in all 4 types we've seen, 2nd level and above of index are always Primary (as index itself is unique and sorted in all 4 cases)

- The rest is for the next tutorial