Mike Medved
mfm190005 - 2882906

**Problem 1. (35 pts)**

Consider a disk with block size $B = 1024$ bytes. A block pointer is $P = 6$ bytes long, and a record pointer is $P_R = 7$ bytes long. A file has $r = 1{,}000{,}000$ EMPLOYEE records of fixed length. Each record has the following fields: NAME (25 bytes), SSN (10 bytes), DEPARTMENTCODE (9 bytes), ADDRESS (35 bytes), BIRTHDATE (8 bytes), JOBCODE (2 bytes), and SALARY (3 bytes).

**a.** Calculate the record size $R$ in bytes.

NAME + SSN + DPTCODE + ADDR + BDAY + JOBCODE + SALARY
25 + 10 + 9 + 35 + 8 + 2 + 3 } 92 bytes for all fields in record

**b.** Calculate the blocking factor $bfr$ (1) and the number of file blocks $b$ (2), assuming an unspanned organization.

(1)
$$bfr = \text{floor}\left(\frac{B}{R}\right) = \text{floor}\left(\frac{1024}{92}\right) = 11$$

(2)
$$b = \text{ceil}\left(\frac{r}{bfr}\right) \approx \text{ceil}\left(\frac{1{,}000{,}000}{11}\right) = 90{,}910$$

**c.** Suppose that the file is ordered by the key field SSN and we want to construct a primary index on SSN. Calculate

(i) the index blocking factor *bfri*

(ii) the number of first-level index entries and the number of first-level index blocks

(iii) the number of levels needed if we make it into a multilevel index

(iv) the total number of blocks required by the multilevel index, and

(v) the number of block accesses needed to search for and retrieve a record from the file—given its SSN value—using the primary index.

(1)
Key field length = 10
Block pointer length = 6
$R_i = 16$
Block size = 1024
$bfri = \text{floor}\left(\frac{B}{R_i}\right) = \text{floor}\left(\frac{1024}{16}\right) = 64$

(2)
number of first level index entries ($r_1$) = 90910
number of first level index blocks ($b_1$) = $\text{ceil}\left(\frac{r_1}{bfri}\right) = \text{ceil}\left(\frac{90910}{64}\right) = 1421$ blocks

(3)
number of second level index entries ($r_2$) = number of first level blocks ($b_1$) = 1421
number of second level index blocks ($b_2$) = $\text{ceil}\left(\frac{r_2}{bfri}\right) = \text{ceil}\left(\frac{1421}{64}\right) = 23$ blocks

number of third level index entries ($r_3$) = number of second level index blocks ($b_2$) = 23 blocks
number of third level index blocks ($b_3$) = $\text{ceil}\left(\frac{b_2}{bfri}\right) = \text{ceil}\left(\frac{23}{64}\right) = 1$ level

Since L3 only has one block, it is the top index level. Thus, index has x=3 levels

(4)
Total number of blocks required by multilevel index = $\sum_{i=1}^{3} b_i$
$= b_1 + b_2 + b_3 = 1421 + 23 + 1$
$= 1445$ blocks

(5)
Number of block accesses needed to search + retrieve record from file given SSN value using primary index
$\Rightarrow x + 1 = 3 + 1 = 4$

**Problem 2. (15 pts)**

Given the same specifications of Problem 1, consider this time you are building a primary index on SSN using B-tree. Calculate (i) the order $p$ for the B-tree, (ii) the number of levels needed if blocks are approximately 69% full (round up for convenience), and (iii) the worst-case number of blocks needed to search for and retrieve a record from the file—given its SSN value—using the B-tree you are estimating.

(1) Order: $p \cdot P + (V+P)\cdot(p-1) \leq$ block size

$P$ = block pointer length, $P = 6$ bytes

$V$ = key field length, $V = 10$ bytes

$\Rightarrow p \cdot 6 + (16)\cdot(p-1) \leq 1024$

$\Rightarrow 6p + (16)(p-1) \leq 1024$

$\Rightarrow 6p + 16p - 16 \leq 1024$

$\Rightarrow 22p - 16 \leq 1024$

$\Rightarrow 22p \leq 1040$

$\Rightarrow p \leq 47.27$

$\Rightarrow p = \text{floor}(47.27) = 47$

Assuming a 69% fill-ratio,

$p = p \cdot 0.69 = 32.43$

$\Rightarrow \text{ceil}(32.43) = 33$

(2) Number of levels needed for p-order w/ 69% fill-ratio

| | Nodes | Pointers | Key Entries | Cumulative Key Entries |
|---|---|---|---|---|
| Root | 1 | 33 | 32 | 32 |
| Level 1 | 33 | 1089 | 1056 | 1088 |
| Level 2 | 1089 | 35937 | 35904 | 36992 |
| Level 3 | 35937 | 1185921 | 1184832 | 1221824 |

Thus, three levels are needed to accomodate 100,000 records.

(3)

Total blocks in 33-order B-tree:

$\sum_{i=0}^{3} \text{blocks}_i \Rightarrow 1+33+1089+35937 = 37060$ blocks

**Problem 3. (30 pts)**

A PARTS file with Part# as key field includes records with the following Part# values: 3, 29, 39, 61, 21, 19, 6, 55, 22, 28, 14, 42, 2, 26, 8, 15, 9, 11. Suppose that the search field values are inserted in the given order in a B+-tree of order $p = 4$ and $p_{leaf} = 3$.

(i) Show how the tree will expand (show all steps as in Fig 17.12 ($7^{th}$ ed)) and what the final tree will look like.

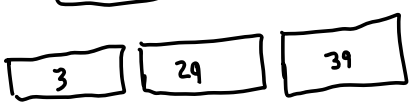
(ii) What is the fill ratio of the B+-tree you created? (Note: we learned 69% is the average fill ratio in class.)

(i) B+Tree Construction:

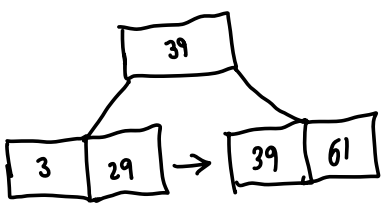
(1)


(2)

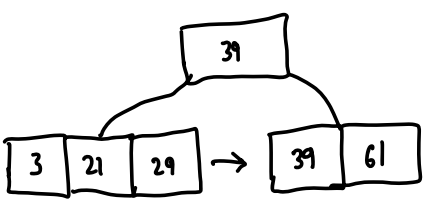

(3)



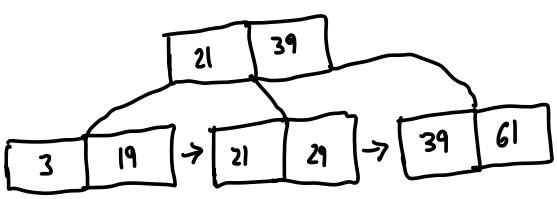

(4)



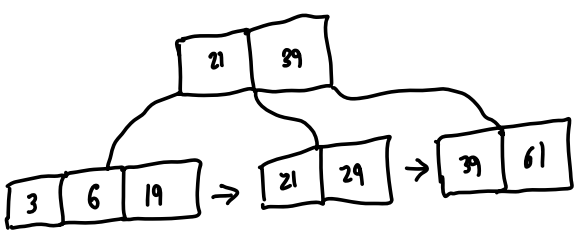

(5)



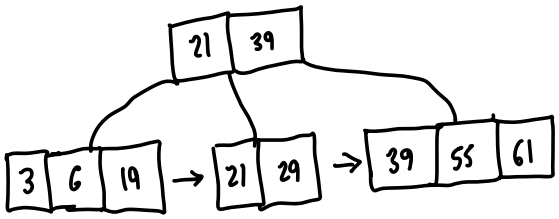

(6)



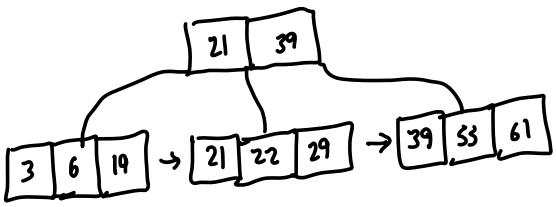

(7)



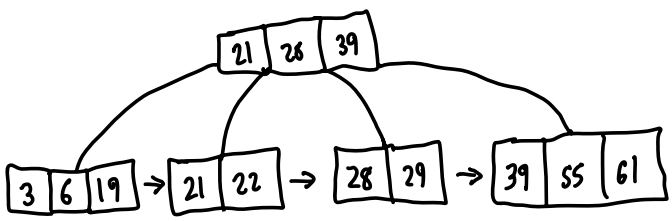
(8)


21 39
3 6 19
21 29
39 55 61

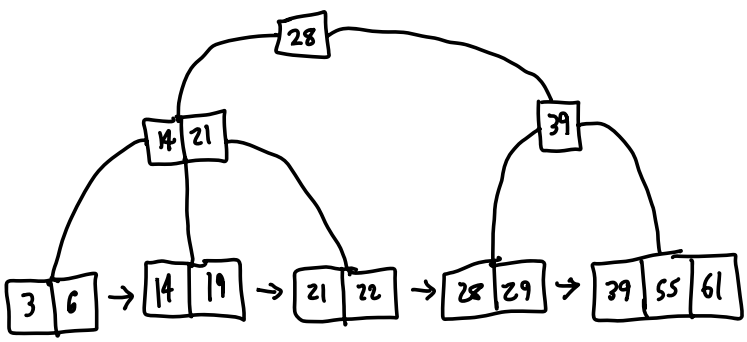
(9)


21 39
3 6 19
21 22 29
39 55 61

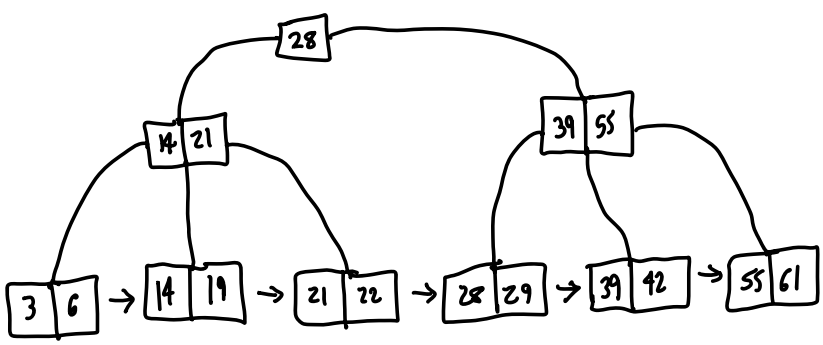
(10)


21 26 39
3 6 19
21 22
28 29
39 55 61

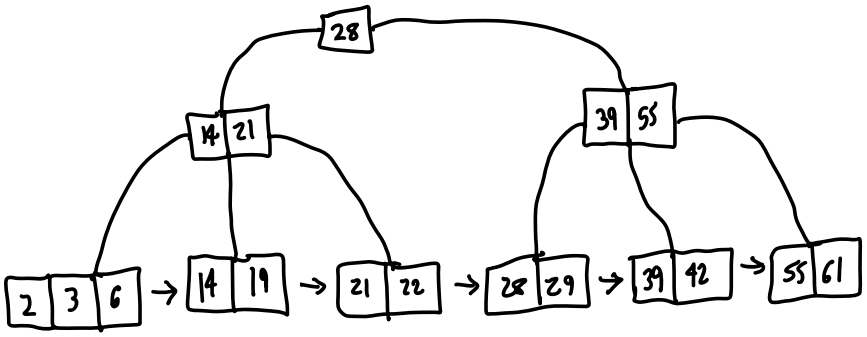
(11)


28
14 21
39
3 6
14 19
21 22
28 29
39 55 61

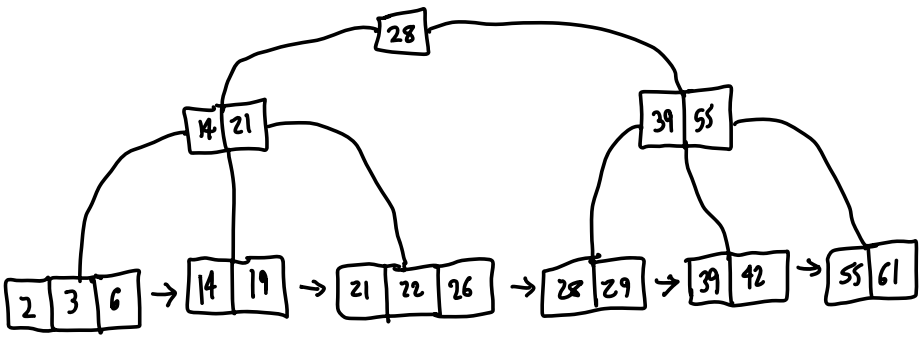
(12)


28
14 21
39 55
3 6
14 19
21 22
28 29
39 42
55 61

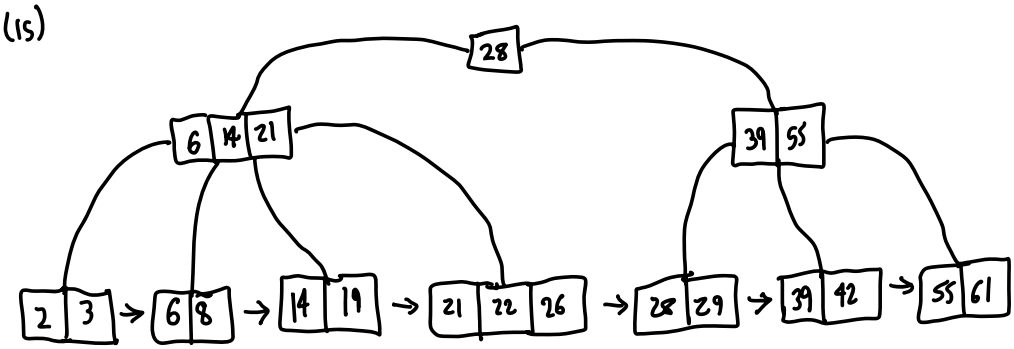
(13)


28
14 21
39 55
2 3 6
14 19
21 22
28 29
39 42
55 61

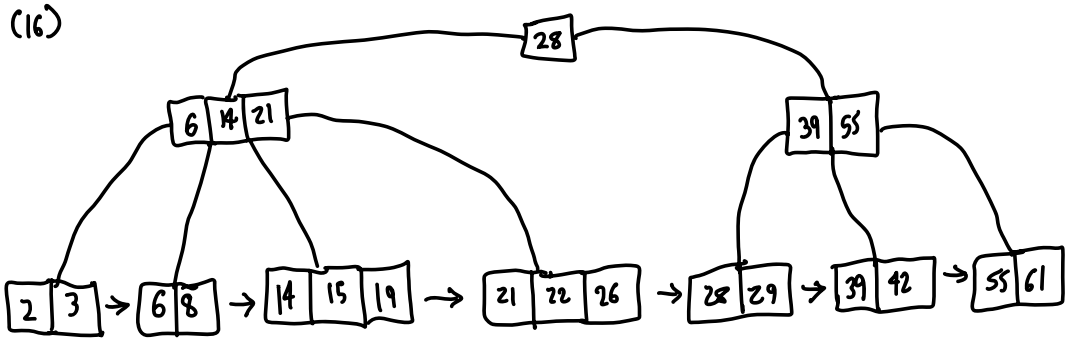
(14)


28
14 21
39 55
2 3 6
14 19
21 22 26
28 29
39 42
55 61

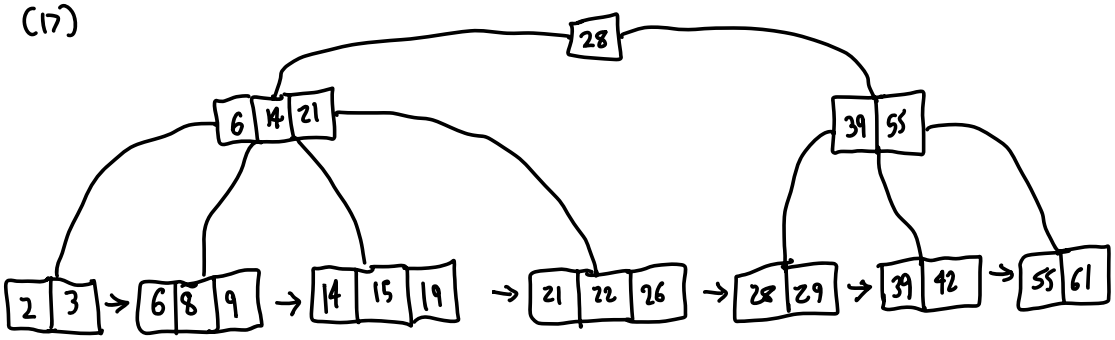
(15)


28
6 14 21
39 55
2 3
6 8
14 19
21 22 26
28 29
39 42
55 61

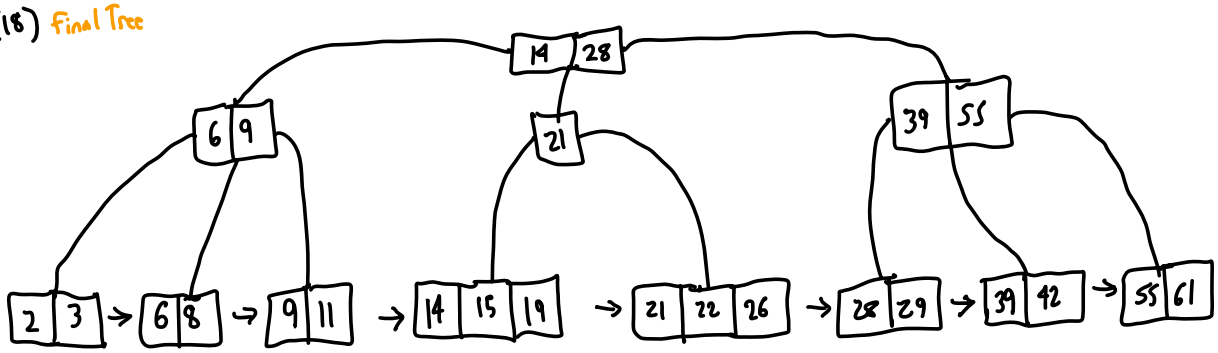
(16)


28
6 14 21
39 55
2 3
6 8
14 15 19
21 22 26
28 29
39 42
55 61

(17)




(18) Final Tree




**(ii) Fill Ratio:**

Only the bottom row = $\frac{18}{24} = 75\%$

All nodes = $\frac{25}{36} = 69.\overline{4}\%$