Review

Exercises

1-) 1000 records that are fixed length and 100 bytes with
- Stored on disk sector size of 102 bytes.
- Each cluster involves 4 sector
- Each track involves 60 sectors.
- 4000 rpm.
- Avg seek time. 60msec.

a) Blocking factor?

$$\left\lfloor \frac{102}{100} \right\rfloor \times 4 = 4 \rightarrow \text{Each block have 4 record}$$

b) Number of block to store the data?

$$\frac{1000}{4} = 250 \text{ block.}$$

c) Avg time to find a record if the read/write head random position initialy and sorted file. (No cylinds based optimal)

Read a one block = Avg seek + Rotational Delay + Transfer

$$6 + \frac{15}{2} + \frac{4}{60} \cdot 15 = 14.5 \text{ ms.}$$

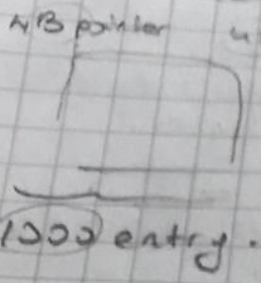
File is sorted → Avg block exist.

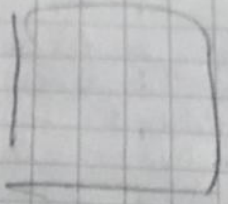
$$\frac{1}{2} \log_2^{250} \cdot 14.5 \text{ ms}$$

d) Time to add new record into the file, Random e/w
File is sorted. Consider worst case.

$$\underbrace{\left(14.5 \cdot \log_2^{250}\right)}_{\text{find a record.}} + \underbrace{\left[250 \times 14.5 \times 2\right]}_{\text{shifting cost}}$$

e) We want to index with dense index. Assume index stores a each key of 4 Byte Pointer 4 Byte search key. How many disk access to find a record?





dense → each record for one entry

1000 record.

Blocksize = ? → Each sector 102
Each record 100
Each block 4 sector.

$\lfloor \frac{102}{8} \rfloor = 12$ ↙ BF=12

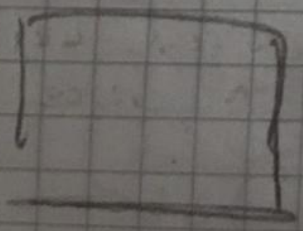
sector

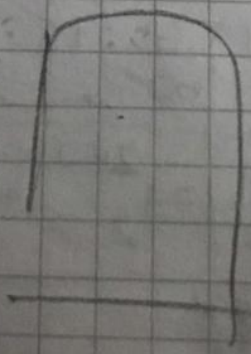
$12 \cdot 4 = 48$

$\lceil \frac{1000}{48} \rceil = 21$

$\log_2^{21} = 5$ disk access.

f) Sparse Index. How many disk access.





Each block for one entry. ↓ 250 Block.

$\lfloor \frac{102}{8} \rfloor = 12$ $(12 \cdot 4 = 48)$ record one block.

$\frac{250}{48} = 6$

$\log_2^6 + 1 = 4$ disk access.

g) Imagine we have (5) buffer pages. Our file is heap file. We want to sort the file. How many runs will be produced in first ...

5-1 = 4 way sorting.

$\frac{250}{5} = 50$ runs in first pass.

$\log_4^{50} + 1 = 4$ pass → total..

total I/O cost =

4 pass * 250 * 2 = 2000 disk accesses

... hashing based

sector size : 102 Bytes
cluster : 4 sectors
track : 60 sectors / 15 cluster
RPM : 4000
Avg seek time: 6 ms

1 sector →
1 cluster →
BF => 4

b) # of blocks
1000 / 4 →

c) Avg time to find a record if the head is at a random position and the file is sorted:

Sorted file we can perform binary search to find a record.

cost of access : seek time + rotational delay = 6ms + 7.5ms = 13.5 ms (her access için geçen süre)

(because head at the random pos. initially)

seek time: 6 ms
rotational delay: 7.5 ms

60000 ms / 4000 rev
? / 1 rev

60000 / 4000 = 15 ms (rotational time)
7.5 ms (rotational delay)

250 blocks
each block has 4 sectors

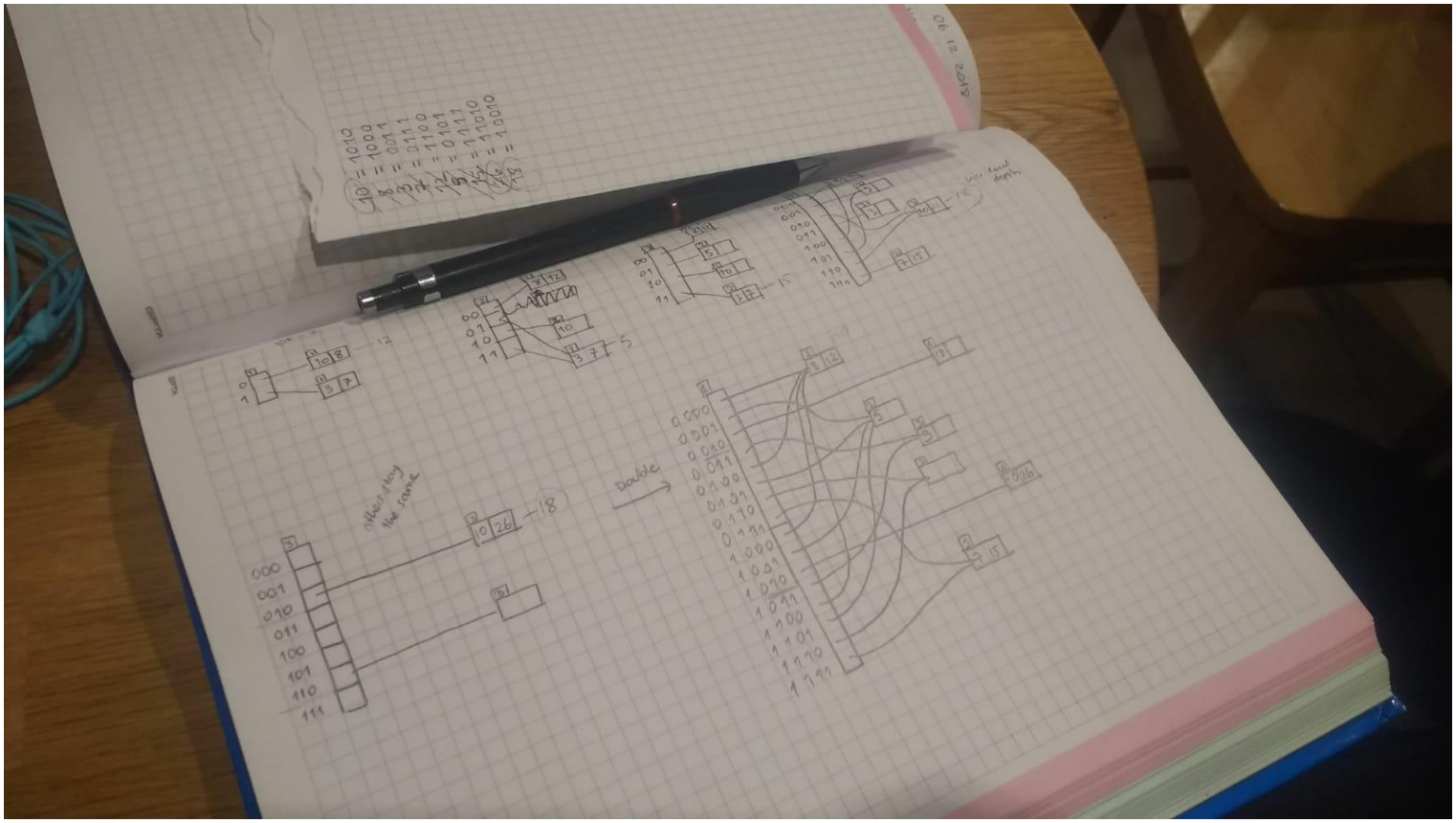
$\lceil \log_2 250 \rceil = \log_2 2^7$

256, 128, 64, 32, 16, 8, 4, 2, 1

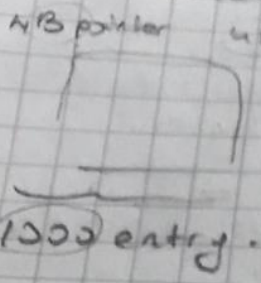
7 disk access

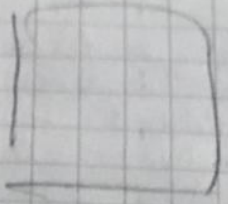
6 ms (seek time) + 7.5 ms * 7 (rotational delay; # of disk access to find a specific record) = 58.5 ms

ya da 7 * 13.5 = 94.5 ms, çünkü hepsi aynı track'te değil


000
001
010
011
100
101
110
111
Double

e) We want to index with dense index. Assume index stores a each key of 4 Byte Pointer 4 Byte search key. How many disk access to find a record?





dense → each record for one entry

1000 record.

Blocksize = ? → Each sector 102
Each record 100
Each block 4 sector.

$\lfloor \frac{102}{8} \rfloor = 12$ ↙ BF=12

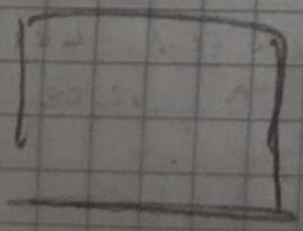
sector

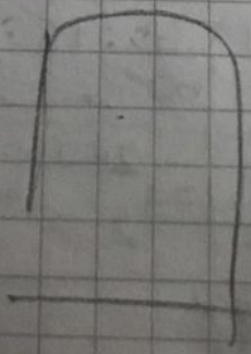
$12 \cdot 4 = 48$

$\lceil \frac{1000}{48} \rceil = 21$

$\log_2^{21} = 5$ disk access.

f) Sparse Index. How many disk access.





Each block for one entry. ↓ 250 Block.

$\lfloor \frac{102}{8} \rfloor = 12$ $(12 \cdot 4 = 48)$ record one block.

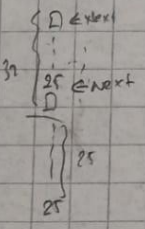
$\frac{250}{48} = 6$

$\log_2^6 + 1 = 4$ disk access.

Question 2 : A database uses linear hashing.
If the Next pointer is pointing to bucket 25!
First bucket is indexed by 0!

a.) What is the min number of primary buckets?

Previous round we have 32 buckets. ($2^4 < 25 < 2^5$)



$32+25=57$

b) How many bucket splits are needed to split the 3rd bucket?

$(32-25)$
$7 + 4 = 11$
$(0,1,2,3)$

c) If Next points to 25, the number of bucket is 89. Which buckets are accessed for searching the following keys?

$H(k_1) = 175$ Round should be 6. 175'in son 6 bitine bakıyoruz 25'ten önceyse bucket + 64 ekleyip o bucketı da kontrol ediyoruz.
$H(k_2) = 83$ 25'ten büyükse kesin oradadır.