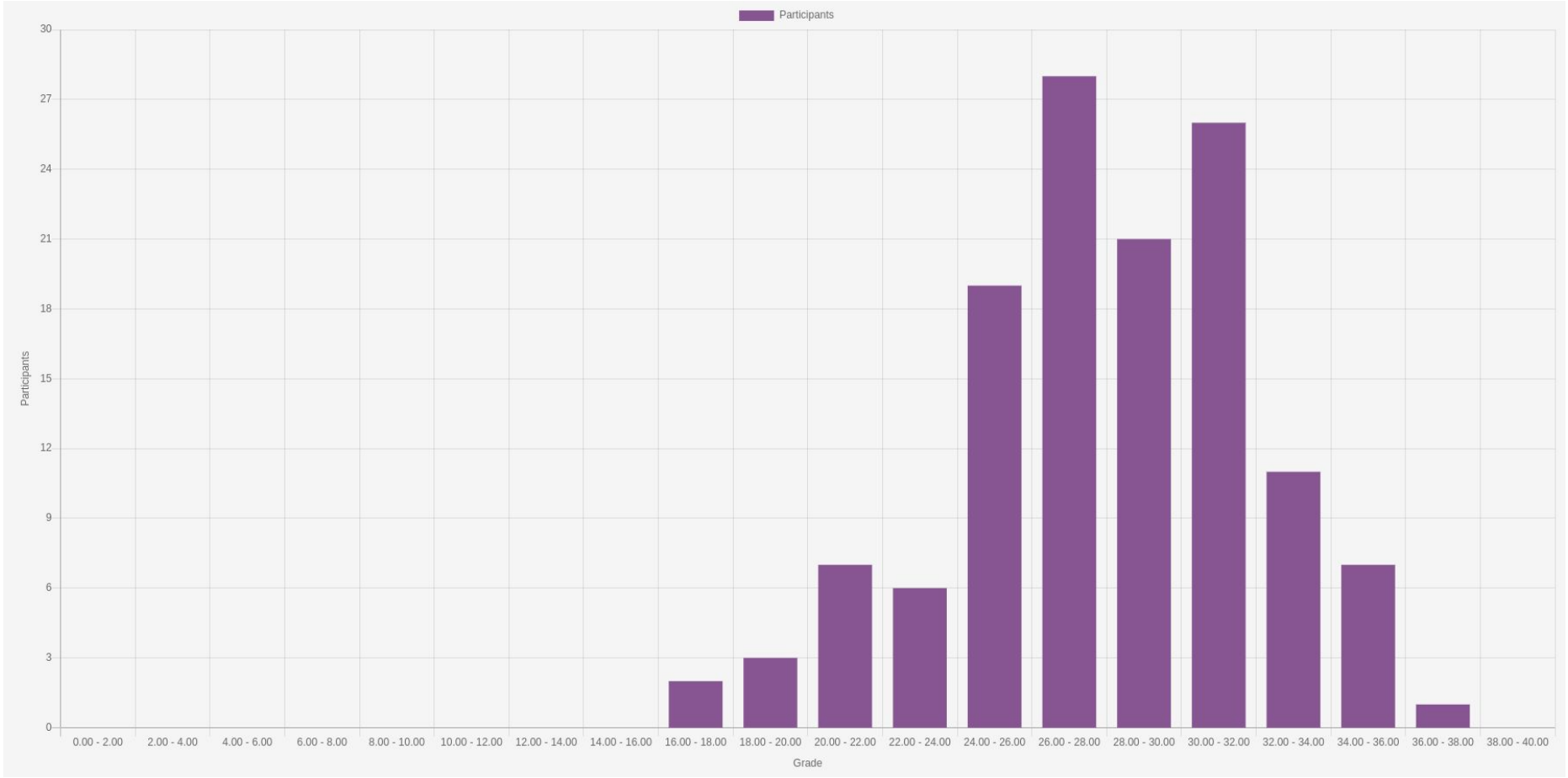
Mid-Term Review

WiSe-22/23


Participants
30
27
24
21
18
15
12
9
6
3
0
Participants
0.00 - 2.00
2.00 - 4.00
4.00 - 6.00
6.00 - 8.00
8.00 - 10.00
10.00 - 12.00
12.00 - 14.00
14.00 - 16.00
16.00 - 18.00
18.00 - 20.00
20.00 - 22.00
22.00 - 24.00
24.00 - 26.00
26.00 - 28.00
28.00 - 30.00
30.00 - 32.00
32.00 - 34.00
34.00 - 36.00
36.00 - 38.00
38.00 - 40.00
Grade

Question 17

Given the following properties of a disk, a clustered data file that is sorted on an attribute a, and a B+ tree index over the attribute a.

disk properties:

- The block size is 10130 bytes.

Data file properties:

- The data file contains 315 million tuples.
- The size of the record representing a tuple is 500 bytes.
- There are no block headers.

Index properties of a dense B+ tree on the attribute a.

- The average fill rate of the nodes (except the root) of the B+ tree is 60%.
- The size of a search key is 2 bytes.
- The size of a pointer is 15 bytes.
- The root (and only the root) of the B+ tree is cached in main memory.

Using the above configuration, please answer the following:

How many blocks are required to be fetched to read 500 records?

How many leaf nodes are there for the B+ tree index?

What is the depth of the B+ tree index?

How many I/O operations does it take to execute a range query on a using the B+ tree that returns 500 records?

Question about all 4 parts of the question. Main concerns about the depth of the tree, it does not match how the question was solved in the quiz. The answers in the quiz depend on this value so some other parts of the questions are also affected.

We agree that the explanation in the exercise do not match with the explanation in the quiz. Therefore, answers for the depth of B+ tree and the number of I/O operations will be updated and accordingly the points will be awarded.

Question 6

Given is a disk with the following characteristics:

- 4 platters with 8 surfaces total.
- 15000 (= 1.5×10^4) tracks per surface.
- 300 (= 3×10^2) sectors per track (this is a simplified model where all tracks have the same number of sectors).
- 2048 (= 2^{11}) bytes per sector.
- 8192 (= 2^{13}) bytes per block.
- 7200 (= 7.2×10^3) rotations per minute.
- 15% of a track gaps between sectors.
- The read/write head can move across 8000 (= 8×10^3) cylinders in 1 ms.
- The read/write head requires 1 ms to start/stop (1 ms for both operations together).

Compute the different timings below.

Average seek time in ms:		(Round to 2 digits after the decimal point.)
Average rotational latency in ms:		(Round to 2 digits after the decimal point.)
Transfer time of a single block in ms:		(Round to 3 digits after the decimal point.)
Random access time for a single block in ms:		(Round to 3 digits after the decimal point.)

There is an issue with the rounding of average seek time that gets cascaded to other answers as well. We will revisit all answers and appropriately award points.

Question 22

Which multi-dimensional index structure primarily partitions data and not space?

Choose an answer:

- Kd-tree
- R tree
- Quad tree

Take an example of this vector/array - [1, 2, 3, 4, 100] and you have to divide this into two sub-vectors based on space and data

Space subdivision - Data space is between 0 and 100. When we subdivide space we select the middle of the data space thus, left child will be less than 50, right child greater than 50, i.e., [1, 2, 3, 4] and [100]

Data subdivision (select middle element from the **data** and subdivide). Middle element is 3 so the subdivision will be [1, 2] and [3, 4, 100]

In Kd-tree you can divide either space or data. In R-tree (think about the question in the quiz for insertion), when you compute the MBRs at each level you take the data of the children nodes (this is **data-driven**). In Quadtree, you take the entire **space** at each level and subdivide it equally into 2k subspaces.

Question 12

RAID levels 1, 4, 5 have the following advantages over RAID 0:

True	False	
○	○	Loses data if a single disk fails
○	○	Strong data consistency
○	○	Increases redundancy
○	○	Faster write and read performance

This question is about advantage of RAID 1, 4, 5 over RAID 0. Students say that all RAID levels apart from 0 have to maintain redundant data and thus that is extra work (both for redundancy and data consistency).

RAID stands for redundant array of inexpensive disks. In this context, redundancy is a desirable property when it comes to keeping mission critical data (banks, stock exchange, etc). Also, the option gives a hint about the same by saying that in contrast to RAID 0 the other RAID levels increases the redundancy. *For strong data consistency we will update the answer to false and re-evaluate the points.*

Question 18

Answer whether the following statement is true or false:

Extensible Hash Table (EHT) allows the number of buckets to double whenever any bucket has too many records.

Select one:

- True
- False

This question is about EHT. We can refer the students to the Complete Book. The statement has been taken from Page 696.

✦ *Extensible Hashing*: This method allows the number of buckets to double whenever any bucket has too many records. It uses an array of pointers to blocks that represent the buckets. To avoid having too many blocks, several buckets can be represented by the same block.

This question is taken from the Complete Book page 696.

Question 7

Given is a table $T(a, b)$ and an index on the attribute a.

The table as the following properties:

- T has 3.6 million (= 3.6×10^6) tuples.
- T takes up 180000 (= 1.8×10^5) blocks on the disk.

The index on attribute a has the following properties:

- The index is fully cached in memory.
- Given a range query on attribute a, the index can be used to determine the number of qualifying tuples and their block IDs.

Both the table and the index are stored on a disk with the following properties.

- Each track can store 600 (= 6×10^2) blocks.
- The average seek time is 9.3 ms.
- The random access time of a single block is 13.6 ms.
- A complete rotation of a track requires 8.33 ms.

The table is not clustered on disk. However, you can assume the following:

- The blocks of the table are stored on as few tracks as possible. (A track that stores tuples of the table is completely filled with tuples from the table.)
- The tracks on which the table is stored are randomly distributed on the disk.

We execute a range query on attribute a that returns 24000 (= 2.4×10^4) tuples.

Scenario (a): The table is **sorted** on the attribute a. How long does it take to retrieve the results of the range query? You can assume that the blocks of the result are stored on as few tracks as possible.

(Round to **closest** integer value.)

Scenario (b): The table is **not sorted** on the attribute a. How long does it take to retrieve the results of the range query?

(Round to **closest** integer value.)

Scenario (c): The table is **not sorted** on the attribute a. However, once we have retrieved the block IDs from the index, we sort them in memory in the order they are stored on disk before accessing the disk. How long does it take **at most** to retrieve the results of the range query?

(Round to **closest** integer value.)

The question is the numerical on access patterns. One person complains that his answers disappeared, while one just says “it is difficult to understand what the question requires”.

Question 13

We have storage configured with 4 hard disks running in RAID 5 configuration. The storage achieves a random write performance of R*N/4 (with N=number of disks and R=Throughput in MB/s for a single disk for random workload) because of the 4 hard disks.

Select one:

- True
- False

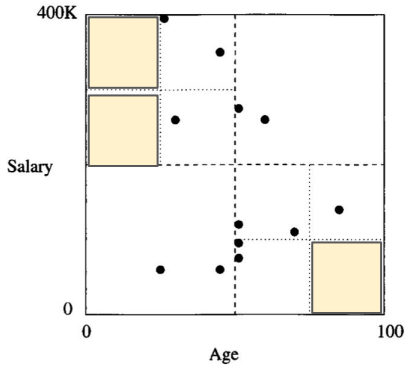
This is about RAID T/F. Complains about too hard to understand this particular variant “We have storage configured with 4 hard disks running in RAID 5 configuration. The storage achieves a random write performance of R*N/4 (with N=number of disks and R=Throughput in MB/s for a single disk for random workload) because of the 4 hard disks.”

The random write performance is R*N/4 because of the operations involved in each write and not because of the number of disks.

Question 24

Which of the following are characteristics of a Quadtree:

True	False	
○	○	A two dimensional Quadtree divides the complete space into four equally sized sub-spaces
○	○	It does not cover the entire space
○	○	It can be sparse
○	○	It is always balanced



Sparse means that some of the leaves can be empty

Question 3

The buffer manager is responsible for speeding up any data access request by caching the pages from the disk into the main memory.

Select one:

- True
- False

This is about this question “The buffer manager is responsible to speed up any data access request by accessing data structures such as indexes.” The student says what about the case when the disk block is not in main memory.

Question 9

PAX aims at storing records/tuples in a row layout on disk and decomposing it into column layout in main memory.

Select one:

- True
- False

In PAX layout, the data is always stored in column layout within each block.